A Structured Approach to Adopting Agile Practices: The Agile Adoption Framework

Ahmed Sidky

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Computer Science and Applications

Approved:

James D. Arthur, Chair

Osman Balci

Shawn Bohner

Manuel A. Pérez-Quiñones

Linda Wallace

May 24, 2007
Blacksburg, Virginia

Keywords: Software Engineering, Agile Development, Agile Adoption, Organizational Change, Agile Readiness, Agile Practices, Agile Potential, Agile Level

A Structured Approach to Adopting Agile Practices: The Agile Adoption Framework

Ahmed Sidky

ABSTRACT

Many organizations aspire to adopt agile processes to take advantage of the numerous benefits that it offers to an organization. Those benefits include, but are not limited to, quicker return on investment, better software quality, and higher customer satisfaction. To date however, there is no structured process (at least that is published in the public domain) that guides organizations in adopting agile practices. To address this situation, we present the Agile Adoption Framework and the innovative approach we have used to implement it. The framework consists of two components: an agile measurement index, and a 4-Stage process, that together guide and assist the agile adoption efforts of organizations. More specifically, the Sidky Agile Measurement Index (SAMI) encompasses five agile levels that are used to identify the agile potential of projects and organizations. The 4-Stage process, on the other hand, helps determine (a) whether or not organizations are ready for agile adoption, and (b) guided by their potential, what set of agile practices can and should be introduced. To help substantiate the "goodness" of the Agile Adoption Framework, we presented it to various members of the agile community, and elicited responses through questionnaires. The results of that substantiation effort are encouraging, and also suggest further avenues for improvement.

DEDICATION

To the people I love the most – my parents.

Dad, the core of this work is named after you: SAMI…

ACKNOWLEDGEMENTS

First and foremost I am grateful and thankful to Allah (God) who blessed me with guidance, good health, and good friends that supported me all through my life. I thank Him for giving me patience and helping me through all the tough times and especially, during the writing of this dissertation. I am also forever grateful to my beloved parents and sisters, who helped me through every step of my life till I have reached where I am today. The Prophet Mohamad (Peace be Upon Him) said: "He who doesn't thank people doesn't thank Allah.[1]" Below I express my thanks and gratitude to the people who have helped me complete this work.

I have been fortunate to work with Dr. James Arthur. Dr. Arthur exemplifies dedication, perfection and professionalism. Throughout his supervision of my work, he did not act as an advisor only. Rather, he was a close friend who stood by me through thick and thin. I will never forget the times we shared together, especially eating Sushi.

I would also like to thank Dr. Osman Balci, Dr. Shawn Bohner, Dr. Manuel Pérez, and Dr. Linda Wallace for serving on my committee. Their help and guidance was vital for the completion of this work. Also, I am grateful to Dr. Sara Thorne-Thomsen for help and advice throughout the writing of this dissertation.

It has been a blessing to work with the Software Engineering Research Group. My colleagues were very supportive and collaborative. They made the lab a pleasant and fun place to study and do research. Special gratitude goes to my dear friend and research partner: Amine Chigani. I would also like to extend my thanks to Ramya Ravichander and Shvetha Soundararajan.

[1] Abu Dawud, Book 41: 4793

I would like to thank all my friends, across the globe, who have supported me through encouragement and prayers. At the same time, I would like to mention a special group of friends who went the extra mile and really sacrificed time or effort for me. These are, Zaki Barzinji, Tamer Desouky, Jamil Renno, Khaled Adjerid, Imad Sheikh, Idris Adjerid, Dr. Mazen Arafeh, Dr. Ehab El Shawarby, Omar Hossino, Abdel Shakoor. Those who I forgot to mention, please forgive me.

Moreover, I am grateful to the wonderful Muslim community in Blacksburg and the United States that had a significant impact on my life. The Muslim community especially the Virginia Tech MSA, VTMUGS and MSA National provided me with moral, spiritual and emotional advice and support and were there for me whenever I needed them.

Last but not least I am grateful to the computer science department, faculty and staff and fellow graduate students who were an essential part of my overall learning experience.

Yet again, all thanks and gratitude goes to Allah only.

(El Hamd el Allah)

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. Introduction

This dissertation presents the Agile Adoption Framework. This framework is a structured approach to guide and assist the process of introducing agile practices into organizations. The present chapter provides a brief history of agile practices and the problems organizations can face when opting to move toward agility. The discussion of the history and problems motivates the creation of the Agile Adoption Framework.

1.1. Background

The work presented in this research is primarily within the area of software engineering. More specifically, this research fits within the area of software process improvement and organizational change. This section presents a brief history of agility starting with the pioneering works up to the current research in the field of agile adoption. We also identify the insufficiency that motivated the creation of the Agile Adoption Framework.

1.1.1 The History of Agility

The main goal of Software Engineering consists of the establishment and use of sound engineering principles and methods to obtain economic software that is reliable and works on real machines [15]. These governing engineering principles and methods form the software development process. After realizing the significance of software development processes in producing "good" software, many efforts arose to identify the "most suitable" software development methodologies. One of the pioneers in this quest was the Software Engineering Institute (SEI). The SEI introduced the Capability Maturity Model for Software (CMM or SW-CMM) to the software development community in 1986. The CMM is a process maturity framework that helps organizations improve their software process through a set of recommended practices in a number of key process areas [40].

With the end of the twentieth century and the beginning of a new millennium, the software market presented new challenges to the software development industry. Some of these challenges are pressure for accelerated product development, minimum time to market, customers demands, and reduced budgets. Traditionalists using the Capability Maturity Model (CMM) and the improved Capability Maturity Model Integration (CMM-I) preferred extensive planning, codified processes and other rigorous means to make development more efficient and predictable. Hence, they were gradually leading the development process towards perfection [19]. Many believed that the traditionalists' approach offered the best solution for the problems of the software industry; but others did not.

Seventeen practitioners sympathetic to finding an alternative to the detailed plan-driven development approach convened in February 2001 [45]. The outcome of this meeting, "The Manifesto for Agile Software Development" [18], declares:

> We are uncovering better ways of developing software by doing it
> and helping others do it. Through this work we have come to value:
>
> **Individuals and interactions** over processes and tools
> **Working software** over comprehensive documentation
> **Customer collaboration** over contract negotiation
> **Responding to change** over following a plan
>
> That is, while there is value in the items on the right, we value the
> items on the left more.

The creation of this manifesto brought to light many different agile development processes and methods and helped many others emerge. A short list of the well known agile methods includes Extreme Programming (XP), Scrum, the Dynamic Systems Development Method (DSDM), Agile Modeling (AM), Adaptive Software Development (ASD), Lean Development (LD), Crystal Methods and Features-Driven Development (FDD). Many others exist.

1.2. The Agile Adoption Wave

At first, organizations were skeptical about adopting agile software methodologies. Information technology (IT) executives were asking the agile community "why should we adopt agile practices?" The agile community provided a tangible hands-on answer exemplified by the numerous pilot projects and small-scale transitions that occurred in organizations. The results were impressive and empirical evidence showed that embracing agile practices yielded many benefits [85], [76], [12], [11], [61], and [59], including:

- Early return on investment
- Short time to market
- Improved quality
- Enhanced client relationships
- Better team morale

The numerous success stories highlighting the benefits reaped by organizations that have successfully adopted agile practices have provided a sufficient answer to those who were doubtful about adopting agile practices. The quick and wide publication of these success stories have caused the agile adoption wave to grow stronger and progress faster.

Many anticipated and noticed the wave of agile adoption. In "Corporate IT Leads the Second Wave of Agile Adoption", a report released on November 30, 2005 Forrester Research noted:

> Agile software development processes are in use at 14% of North American and European enterprises, and another 19% of enterprises are either interested in adopting Agile or already planning to do so. Early adopters of agile processes were primarily small high-tech product companies. But a second wave of adoption is now underway, with enterprise IT shops taking the lead. These

> shops are turning to agile processes to cut time-to-market, improve quality, and strengthen their relationships with business stakeholders.

Other surveys and published research, such as the *Agile 2006* survey by Digital Focus, the *State of Agile Development* survey conducted by VersionOne, and the *Agile Adoption Rate* survey conducted by Scott Ambler, all show evidence that agile adoption is on the rise.

1.3. Motivation for the Agile Adoption Framework

With the growth of agile adoption, the question of *why* to adopt agile practices was quickly being replaced with *how* to adopt agile practices [46]. The tone of those interested in Agile shifted from asking about examples where agility works, or does not, to asking for guidance and assistance on how to implement agility in their particular cases.

Although it was possible to point to numerous successful agile adoptions, these success stories cannot be generalized. Many of them were too narrowly focused at a specific organization. These success stories cannot provide useful guidance and assistance to another organization with a different set of needs. However, the mere presence of such success stories triggered people to ask if the same approach would work at their organization. Thus, the need for a set of generic guidelines to help identify the right agile practices for an organization considering moving to agility became evident. These guidelines also needed to warn of possible pitfalls in adopting certain agile practices, outline the order in which practices should be adopted and provide answers to many other questions related to agile adoption.

Even though consultants or agile coaches could provide answers to many of these questions, it is important to realize that such answers were based on their own experiences. In many cases, the knowledge of agile coaches was limited to the type and size of projects they were involved with. Typically, after the completion of a project, consultants reflect on what worked and what failed. They would then retune their adoption approach for their next project. Some of them had even captured these skills in publications and have suggested to the agile community some successful practices, or what to be careful of when adopting agility [71, 46, 67, 19]. These contributions are valuable and do provide those seeking to adopt agile with some level of guidance and assistance. However, this advice is not structured in a structured approach of the sort really needed to guide efforts of organizations in adopting agile practices.

Like consultants, researchers have developed a limited number of approaches to help and guide people with agile adoption. For example, Bohem designed a framework to guide agile adoption efforts based on the assessment of various risks. Bohem's framework is very helpful as it provides those seeking to adopt agile with a way to carefully balance agility and discipline. While Bohem's work is helpful, one of its drawbacks related to agile adoption is that it addresses agility in its generic form, instead of focusing on actual practices. Organizations seeking to adopt agility still need some tangible approach. They are still required to identify the actual agile practices that are the best fit for their organization before attempting the move to agility.

With the agile adoption wave growing, and with the absence of structured and disciplined guidance and assistance from the agility community, there is a need for a new approach to help address the concerns surrounding *how to adopt agile practices*.

1.4. Problem Statement

This research addresses the current absence, at least in the public domain, of a structured approach to guide agile adoption efforts. Furthermore, a rigorous formalization of what constitutes agility is also missing. Organizations aspiring to become agile want to know when they are considered "agile," as well as what it means to be "agile". Moreover, guidelines highlighting what is needed to help agile adoption efforts succeed are unavailable. These guidelines are essential for determining if any activities or tasks are overlooked during the adoption process. The lack of a structured approach for agile adoption causes organizations to question how to identify the right practices to adopt, how to determine if they are ready for agile, what the necessary preparations for agile are, and what the potential difficulties that could develop during the adoption process are.

The contributions consultants and researchers have made are valuable, and have developed the foundations for guiding agile adoption efforts. However, the lack of structure in some approaches and the lack of focus on agile practices (a problem Bohem's approach exemplifies) prevent these contributions from ultimately providing structured guidance and assistance to organizations on how to adopt agile practices. Furthermore, because of the narrow focus of these approaches, even when they provide some guidance, it is not comprehensive enough, because it only treats a small aspect of the adoption effort or draws on the process used at a single organization with specific parameters and expectations. Therefore, what the agile industry needs is a set of repeatable guidelines (a framework) that agile coaches can use to determine an organization's readiness for embarking on the journey toward agility and to guide the actual adoption process.

To develop a framework that can provide organizations with comprehensive guidance and structured assistance for their agile adoption journey, a number of issues have to be addressed. Among the most important and challenging of these issues are how to:

- introduce structure in a complex and unpredictable process like that of agile adoption
- measure and assess agility independent of agile methods
- accommodate project and organizational characteristics influencing agile adoption efforts

- ensure that the framework guides the adoption effort in an efficient and effective manner.

The next sections briefly elaborate on each of these issues.

1.4.1 Introducing Structure into the Agile Adoption Process

Transitioning organizations to agility is an unpredictable process. This is mainly due to several aspects of the organization including, but not limited to, its structure, people, culture, and management practices. All of these features can affect the effort at any point in the journey to adopting agility. Hence, a key issue that arises when developing a framework to guide and assist with adoption efforts is how to encompass the efforts of such an organization-wide change phenomenon within a comprehensive structured framework. Some of the many questions arising from this issue are:

- how to develop a framework that provides enough structure to guide and assist agile adoption efforts while not dictating them
- how the framework can determine the agile practices most suitable for the organization to adopt
- how to capture the order in which agile practices should be adopted in the organization
- how the framework should handle situations where the organization is not ready for certain agile practices.

1.4.2 Measuring and Assessing Agility

Measurement and assessment are substantial components of many process improvement efforts, including the move to agility. Therefore, a measurement approach is needed when analyzing the current agility of the process and identifying its agile potential. The main challenge is how to measure or assess agility for a project or organization independent of a particular agile methodology. Some of the factors contributing to this challenge are:

- identifying a suitable measurement scale for agility
- determining the aspects of the development process that need to be assessed to conclude its agility
- finding a way to aggregate the assessment results of all these different aspects in a manner that enables the assessor to determine the agility of the project or organization.

1.4.3 Accommodating Project and Organizational Characteristics

Agile processes accommodate and adapt to different situations and environments, and the transition to agility should be no different. Although the framework guiding agile adoption needs to be repeatable, it also needs to accommodate the changing nature of projects within an organization, just as agile processes accommodate the changing nature of requirements in a project. Each software development project is unique and surrounded by unique characteristics. In light of this, another challenging issue is how to develop a framework that can accommodate the unique characteristics of each project within the organization. Some of the challenges related to this issue are:

- addressing adoption efforts related to individual projects without overlooking the overall organization
- how to differentiate between, and handle, project characteristics that can be changed, as well as those that cannot because they are outside the project's and/or organization's control.
- dealing with scenarios where a project's ability to adopt agile practices is different from an organization's ability to adopt agile practices.

1.4.4 Effectively Guiding the Agile Adoption Effort

Effectiveness is a key principle of agile development processes. To uphold this principle when transitioning organizations to agility brings about another challenge: how to ensure that the framework is guiding the transition to agility in an effective manner. Ensuring that such a complex process (i.e. agile adoption) is conducted effectively is challenging due to a number of factors, including:

- determining if the organization is ready for transitioning to agility *before* committing any resources to the adoption effort. It is clearly ineffective to waste time, effort and money on trying to transition an organization that is not ready for agility.
- Ensuring before starting the actual adoption process for a particular practice that the organization is ready to adopt that practice; again committing time, effort and money trying to introduce a practice into an organization that is not ready for the practice is ineffective
- designing the framework in a way that it assesses and measures only the aspects of the organization that are necessary and sufficient, and avoids any extra efforts that do not directly contribute to the adoption effort.

1.5. Solution Approach

The Agile Adoption Framework presented in this dissertation is a structured and efficient approach to guide agile adoption efforts within projects without overlooking the organizational aspect of the adoption process. The Agile Adoption Framework tackles the four issues mentioned in the previous section through its unique design and structure. The framework has two main components: the Sidky Agile Measurement Index (SAMI) and a 4-Stage process that utilizes SAMI to determine which, and to what extent, agile practices can be introduced into the organization.

The first component, SAMI, serves three important purposes.

- It serves as a tool to measure and assess the agile potential of an organization independent of any particular agile method (e.g. XP, Scrum …etc)

- It provides a scale for identifying the target agile level for a project aspiring to adopt agility.
- The measurement index helps the coach organize and group the agile practices in a structured manner based on essential agile qualities and business values.
- It provides a hierarchy of measurable indicators used to determine the agility of an organization.

However, SAMI by itself does not guide organizations adopting agile practices. The second component of the Agile Adoption Framework, the 4-Stage process, uses SAMI to guide organizations by identifying the agile practices that are most suitable for their environment. Each of the four stages within this component of the framework (4-Stage process) are carefully sequenced and designed to ensure that the adoption process is conducted in a highly effective manner. The four stages are:

- Stage 1: Identification of Discontinuing Factors
- Stage 2: Project Level Assessment
- Stage 3: Organizational Readiness Assessment
- Stage 4: Reconciliation

The 4-Stage process introduces the structure into the agile adoption process because each stage has clearly defined inputs, outputs and objectives. The next sections elaborate briefly on each of the four stages while showing how they help provide structured and efficient guidance and assistance to organizations adopting agile practices.

1.5.1 Stage 1: Identification of Discontinuing Factors

Since adopting agile practices is essentially a type of Software Process Improvement (SPI), the organization needs to undergo a pre-assessment phase before it makes the decision to start the initiative. Therefore, the objective of this first stage is to identify whether an organization is capable of embarking on the journey of transitioning to agility. This is accomplished by providing organizations with a means to decide whether or not to proceed with agile adoption initiatives. Conducting such an assessment before any effort is put into the adoption initiative is efficient because it saves the organization from committing valuable time and resources to a SPI initiative that is destined to fail. This stage is completed when either a “Go or No-go” decision made at the beginning of the agile adoption effort.

1.5.2 Stage 2: Project Level Assessment

Stage 2 starts once a Go decision is made from Stage 1. The main objective of this stage is to identify the highest level of agility that a *project* can achieve. Stage 2 has a project-level focus because each project in an organization is unique. This implies that each project in the organization can function at different levels of agility. Therefore, in this stage, the focus is on identifying and assessing the existence of project-level factors, especially those outside the organization’s control, that have the potential to jeopardize the success of the adoption of agile practices. This assessment process is accomplished by utilizing the assessment indicators identified within SAMI. This stage ends once a target agile level (from SAMI) is identified for a project aspiring to adopt agile practices.

1.5.3 Stage 3: Organizational Readiness Assessment

The input needed for Stage 3 is the project’s target agile level, which is determined in Stage 2. The objective of Stage 3 is to determine the extent to which the organization is ready to support the adoption of the project’s target agile level. To determine this, an assessment is conducted, using SAMI, that looks at how accommodating the different components of the organization (i.e. developers, tools, culture…etc) are for the adoption of the agile practices contained within the project’s target agile level. By identifying what each agile practice needs for its successful adoption, and then determining whether the organization meets these needs, the organization is able to discover the necessary preparations it needs to undertake to adopt the desired agile practices. The output of this stage is the level of agility (from SAMI) the organization is ready to adopt.

1.5.4 Stage 4: Reconciliation

The Reconciliation stage begins after a target agile level for the project is identified (from Stage 2) and an organizational readiness agile level is determined (from Stage 3). The objective of this stage is to reconcile any difference that may exists between these two levels. During this stage, the differences between the practices the project *wants* to adopt (i.e. the project's target agile level) and the practices the organization *can* adopt (i.e. organization's readiness level) are resolved. By resolving these differences (if any), this stage results in determining the set of agile practices that are most suitable for the organization to adopt.

The Agile Adoption Framework assists the agile community in supporting the growing demand from organizations that want to adopt agile practices. However, the framework is only one essential ingredient; the other is an agile coach who knows how to apply that framework. Such a person can be an agile consultant hired to facilitate the process, or an in-house employee with sufficient training in agile methods and the use of the framework. The remainder of this dissertation presents in detail how the different components of the Agile Adoption Framework are designed and utilized to provide guidance and assistance to organizations on *how* to transition to a more agile software development process.

1.6. Organization of this Dissertation

The next chapter, Chapter 2, discusses the notion of Software Process Improvement (SPI) and its relation to the Agile Adoption Framework. Chapter 3 presents the structure and details of the Sidky Agile Measurement Index (SAMI). Each of the stages in the 4-Stage process is then presented in detail in Chapter 4. Chapter 5 presents industry feedback regarding the framework. Finally, Chapter 6 provides concluding remarks about the Agile Adoption Framework.

2. Process Improvement and the Agile Adoption Framework

Since adopting agile practices is essentially a process improvement effort, it is useful to the understanding of the components of the Agile Adoption Framework to discuss some generic process improvement frameworks and models. This chapter, therefore, focuses on the concept of software process improvement and its relation to adopting agile practices.

This chapter also puts the Agile Adoption Framework in perspective in terms of overall process improvement efforts. Edward Deming, one of the pioneers of applying statistical process control in industry, has described process improvement as a continuous cycle of six steps [83]:

1. Understand the status of the development process
2. Develop a vision of the desired process
3. List improvement actions in order of priority
4. Generate a plan to accomplish the required actions
5. Commit the resources to execute the plan
6. Start over at step 1

Deming's steps essentially provide a skeleton for process improvement efforts. Deming is even more famous for the cycle he created. Shewart-Deming's cycle consists of four steps (Plan, Do, Study, Act) that capture the same principles as his six steps, but at a higher and more generic level. Shewart-Deming's Cycle can be applied to process improvement efforts in any domain.

Instead of building the discussion in this chapter around Shewart-Deming's Cycle, the focus is on a more detailed model (influenced by Deming) that was originally developed for Software Process Improvement (SPI) efforts. A number of different improvement models are used to support continuous, top-down SPI in organizations. Some of these are the Quality Improvement Paradigm (QIP) [14], IDEAL (acronym for Initiating, Diagnosing, Establishing, Acting and Learning) [66], and ISO/IEC 15504 Part 7 [52]. Table 1 shows the different phases of all the mentioned organizational process improvement models and their relation to the Shewart-Deming Cycle [74].

Shewart-Deming Cycle	QIP	IDEAL	ISO/IEC 15504
Plan	Characterize and understand	Initiating	Examine organization's Needs
	Set goals	Diagnosing	Initiate process improvement
	Choose processes, methods, techniques and tools	Establishing	
Do	Execute	Acting	Prepare and conduct process assessment
			Analyze results and derive action plan
			Implement improvements
Check			Confirm improvements
Act	Analyze	Learning	Sustain improvement gains
	Package and store experience		Monitor performance

Table 1. Organizational Process Improvement Models

The IDEAL and ISO/IEC 15504 models are the best candidates to use to discuss agile process improvement efforts. However, due to the similarities between the models, instead of focusing on two SPI organizational models, IDEAL will serve as the reference model because it is more widely used and claims seniority in terms of age over the ISO/IEC 15504 model.

2.1. The IDEAL Model

At the mention of SPI, many immediately think of SPI frameworks such as Capability Maturity Model Integration (CMMI), ISO 15504 (SPICE), ISO 9001 and so forth. However, this almost automatic response raises the question of where do these frameworks fit in the overall process improvement effort of an organization. For example, CMMI identifies the steps along the path of process maturity, but does not treat how to sustain the progress or provide the context for these changes. The IDEAL process model fills this gap by providing an organizational model for software process improvement. In other words, it provides an overarching model for CMMI and the other process improvement frameworks. The IDEAL model is cyclical, which implies continuous process improvement. That is, it achieves regular and continuous improvement by continuing to go through the cycle of initiating, diagnosing, establishing, acting and learning.

2.1.1 Overview of the IDEAL Model

The IDEAL model provides a disciplined engineering approach for improvement. It focuses on managing the improvement program, and establishes the foundation for a long-term improvement strategy. The model consists of five phases:

- **I – Initiating:** Laying the groundwork for a successful improvement effort.
- **D – Diagnosing:** Determining the present state and desired state and developing recommendations for improvement.
- **E – Establishing:** Planning the specifics of how to reach SPI initiative’s target.
- **A – Acting:** Doing the work according to the plan.
- **L – Learning:** Learning from the experience and improving the ability to adopt new technologies in the future.

Each of the five phases is made up of several activities. Figure 1 shows the phases and activities within each of the phases of IDEAL.

Usually in SPI efforts, a team of specially trained individuals, headed by an experienced *change agent,* oversees and conducts the execution of the overall organizational change and improvement process. The phases guiding this team are presented in greater detail:

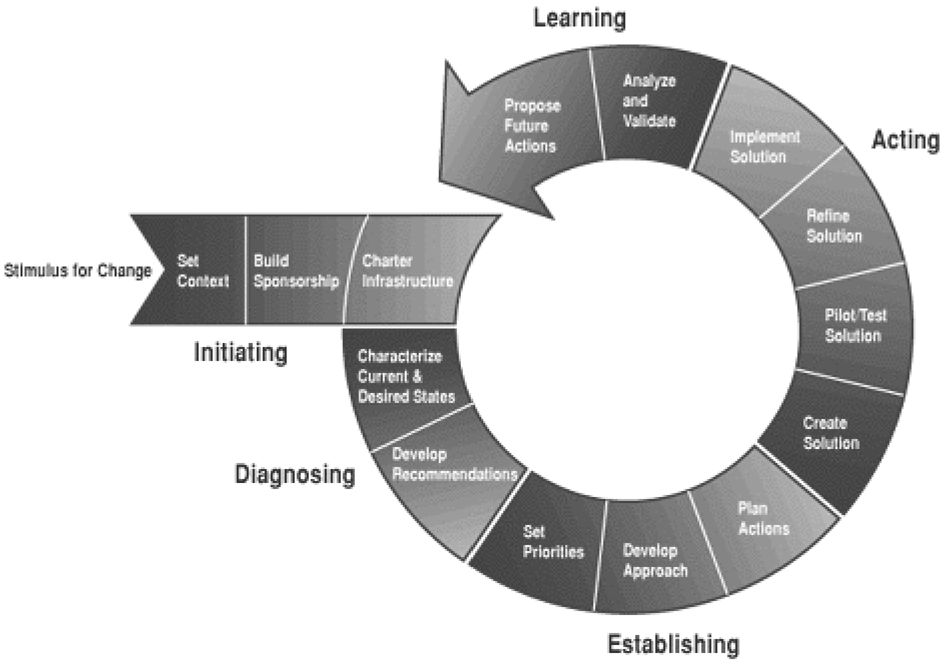


Figure 1. IDEAL Model [66]

- **The Initiating Phase:** The team completes the critical groundwork during the initiating phase. It clearly articulates the business reasons for undertaking the process improvement effort. It identifies the effort's contributions to the goals and objectives of the business. The team secures the support of critical managers, and arranges for the allocation of resources. Finally, it puts in place an infrastructure for managing implementation details. The activities of the initiating phase are critical. If they are done completely and properly, subsequent activities can proceed with minimal disruption. If they are done poorly, incompletely, or haphazardly, then time, effort and resources will be wasted in subsequent phases.

- **The Diagnosing Phase:** This phase builds upon the initiating phase to develop a more complete understanding of the improvement work. During the diagnosing phase, the SPI team determines two characterizations of the organization: the current state of the organization and the desired future state. It uses these organizational states to develop an approach for improving the practice of the business. Characterizing the current and desired states is similar to identifying the origin and destination of a journey. The team can characterize these two states more easily by using a reference standard (or measurement index) such as the Capability Maturity Model (CMM). The recommendations the team develops as a part of this activity suggest a way of proceeding in subsequent activities.

- **The Establishing Phase:** The purpose of the establishing phase is to develop a detailed work plan. The SPI team sets the recommendations made during the diagnosing phase, as well as the organization's broader operations and the constraints of its operating environment. Then, the SPI team develops an approach that honors and factors in the priorities. Finally, the team incorporates the specific actions, milestones, deliverables and responsibilities into an action plan.

- **The Acting Phase:** The activities of the acting phase help an organization implement the work that the SPI team has conceptualized and planned in the previous three phases. These activities typically consume more calendar time and more resources than all of the other phases combined.

- **The Learning Phase:** The learning phase completes the improvement cycle. One of the goals of the IDEAL Model is to continuously improve the ability to implement change. In the learning phase, the SPI team reviews the entire

IDEAL experience to determine what has been accomplished, whether the effort achieved the intended goals, and how the organization can implement change more effectively and/or efficiently in the future. The SPI team must keep records throughout the IDEAL cycle with this phase in mind.

In his thesis titled "Development and Evaluation of Software Process Improvement Methods" Komi-Sirviö provided a detailed discussion of SPI models and frameworks that complement this brief introduction to IDEAL [58].

Before discussing agility and process improvement, it is important to explain where CMMI and the Standard CMMI Appraisal Method for Process Improvement (SCAMPI) fit within the IDEAL model, because later it helps explain where the Agile Adoption Framework fits within the SPI initiatives for agile software development.

2.1.2 IDEAL, SCAMPI and CMMI

The CMMI is the basic measurement index for software development capability and maturity in an organization. The SCAMPI appraisal method is used to identify strengths, weaknesses and ratings relative to the CMMI measurement index. Diagnoses from SCAMPI appraisals should be part of a process improvement cycle such as IDEAL. In the IDEAL cycle, the SPI team uses SCAMPI appraisals and the CMMI primarily in the diagnosing phase to characterize the current and desired states of the organization, and to develop the process improvement recommendations. Figure 2 shows how the CMMI and the SCAMPI are used in this phase.

Additionally, the SCAMPI and CMMI indirectly affect the activities of the Establishing stage of IDEAL. This stage focuses on prioritizing the changes that the organization needs to make and developing an action plan for the process improvement effort. The sequencing of the CMMI levels along with the appraisal approach the SCAMPI uses implicitly dictates a certain prioritization of these process improvement changes. For example, an organization at CMM Level 2 gives a higher priority to establishing the key process areas highlighted in CMM Level 3 than those in CMM Level 5. The purpose CMMI and SCAMPI serve in the establishing phase is to provide direction to the activities in this phase, not to directly conduct these activities. Consequently, they are not officially considered part of the stage.

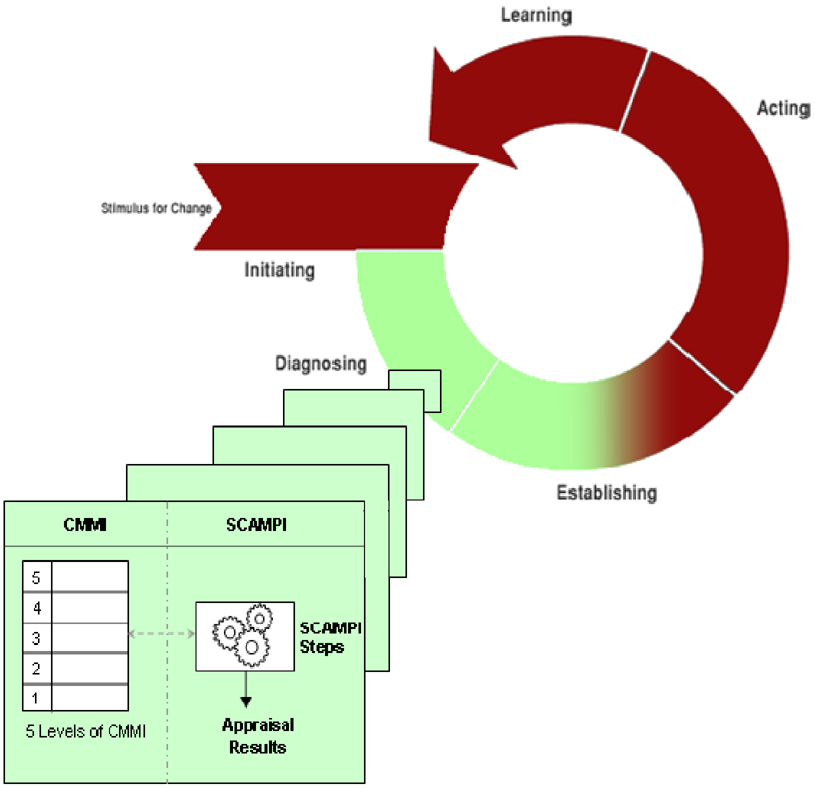


Figure 2. CMMI and SCAMPI relative to IDEAL

The presentation of a generic SPI life-cycle model (IDEAL) and the relationship between it and other process improvement methods and frameworks (SCAMPI and CMMI) leads to further discussion of the SPI life-cycle model, but from the perspective of adopting agile practices in the next section. The next section also includes a discussion of the relationship between the Agile Adoption Framework and SPI models.

2.2. SPI Life-Cycle Models for Agile Development

Moving an organization toward having an agile development process through the adoption of agile practices is a type of process improvement effort. Organizations are increasingly recognizing the need for specific implementation guidance when they adopt new software engineering tools, processes and methods. Because many software process improvement efforts, including the adoption of agile practices, are complex with effects far reaching in the organization, they require a systematic approach for managing the process improvement life-cycle.

While many agile practitioners are critical of CMM and anything related to CMM (e.g. IDEAL), there is no doubt that there is much to learn from the CMM-based process improvement efforts. One important lesson highlights the need to approach process improvement efforts in a systematic manner, instead of chaotically. In fact, some level of guidance and structure should exist for all process improvement efforts, including agile software development, no matter what the anticipated outcome is.

2.2.1 Challenges with SPI Models for Agility

There are several challenges to overcoming the perceived incompatibility of SPI models and agile software development. One challenge is the limited number of references directly addressing the issue of organizational SPI within the context of agile software development. A reason for this might be that most of these SPI models were not originally developed for agile processes. For example, since McFeeley originally designed IDEAL as a life cycle model for software process improvement based upon the Software-CMM (SW-CMM) [66], many agile consultants have perceived it to be incompatible with agile approaches. Their rationale behind that perception is the infeasibility to have plan-driven process improvement models used when the target development method is agile. This reasoning is invalid, because the modifications made to these models, as they exist today, to fit a wide range of process improvement domains suggest they can be modified to fit agile adoption efforts. This leads to the second reason for the claim that SPI models are not applicable to agile adoption efforts. The focus of numerous agile methodologies is on the project level activities of software development, but SPI models address issues on an organizational level [55].

This difference in focus between agile methodologies and SPI models is a valid concern for not using traditional SPI models with agile software development. However, SPI models such as IDEAL provide enough guidance and benefit to make it irrational to throw them all away, because they focus on an organizational level rather than a project level. The Agile Adoption Framework is not a SPI life-cycle for agile development; nor is the focus of this research to provide a complete SPI life-cycle for agile software development. The Agile Adoption Framework, like SCAMPI and CMMI, needs to live within a SPI life-cycle model. The Agile Adoption Framework can be used with a SPI life-cycle model such as IDEAL or ISO/IEC 15504 when the model is slightly modified. These modifications change the initial stages of the model to both cater to the project and organizational levels of the agile adoption effort.

Therefore, since the Agile Adoption Framework can alter IDEAL to include a project focus, it is still possible to use IDEAL as an overall guidance model for agile SPI initiatives.

2.2.2 IDEAL and Agility

In general, IDEAL or similar SPI models (e.g. ISO/IEC 15504) do not demand or provide any specific SPI methods for conducting different SPI activities. Instead, they suggest the steps necessary to achieve continuous improvement, thus serving as a roadmap for improvement in organizations. This section provides a brief look at the steps and phases of IDEAL with the idea of using this model to guide an organization adopting agile practices. Again, the intent of this research is not to modify IDEAL for use with agile adoption, but to present it to provide the context and establish the overall picture of where the Agile Adoption Framework lies within the overall process of SPI.

The SPI initiative using IDEAL starts with a stimulus for change. In the case of agile process improvement, this stimulus usually comes from an organization's aspiration to become more able to adapt to change or to increase product quality or to have an earlier return on investment or any of the other benefits that the adoption of agile practices can realize for an organization [85], [76], [12], [11], [61], [59]. Once a stimulus for change occurs, the IDEAL life-cycle begins.

In the initiating phase, the SPI team still needs to build the context of this desire to move towards agility within the organization. The team is also required to align this transition with the organization's business strategy and build sponsorship for the change. Others have mentioned the same steps highlighted in the initiating phase of IDEAL as "best practices" when adopting agile practices [19, 31]. Therefore, simply following a well established model like IDEAL for process improvement can save the agile adoption industry both time and effort sorting out what works from what does not work for an organization. Of course, this is no silver bullet, but it does save the agile adoption industry from re-inventing the wheel or "learning from scratch."

Once the initiating phase provides the ground work for the move toward agility, the SPI team moves to the diagnosing phase to assess and further analyze the current agile state of the organization until it can develop a set of recommendations concerning the agile practices the organization should adopt. Working from these recommendations in the Establishing stage, the SPI team develops a detailed work plan that takes into consideration the current environment and constraints of the organization. With this information, the SPI team implements the process improvement efforts in the Acting stage. The Learning stage consists of reviewing the changes and reflecting on the change process. If the review shows that the objective of the improvement has not been attained, another cycle of process improvement that takes into consideration what was learned from the previous cycle commences.

Once again, it is important to keep in mind that the SPI team simply uses IDEAL here to provide guidance to agile adoption efforts and not to dictate them.

2.3. Research Scope: The Agile Adoption Framework

Since the scope of this research does not include developing or tailoring a complete process improvement lifecycle for agile adoption efforts, the discussion of IDEAL fulfills the objective of creating the groundwork or foundation for such a lifecycle. Like IDEAL, which needs the CMMI and SCAMPI for its foundations, an agile process improvement lifecycle needs an agile measurement index and a process or method that uses this measurement index to diagnose the organization's readiness and to guide the process improvement effort. This measurement index and the ensuing process are the scope of this research. The Agile Adoption Framework consists of two components, the Sidky Agile Measurement Index (SAMI) and a 4-Stage process that uses SAMI. These two components work together to provide a first step in guiding organizations toward adopting agile practices.

As developed, these two components enable others to start using a process improvement lifecycle, such as IDEAL, to guide their adoption efforts. The challenges, as highlighted in Chapter 1, are to create an agile measurement index and develop a process that efficiently uses this measurement index to guide the adoption efforts. Another major challenge, as mentioned earlier, is to create the 4-

stage process of the Agile Adoption Framework in a manner that addresses project level factors of agile adoption as well as organizational factors.

The mapping of the Agile Adoption Framework to IDEAL primarily covers the activities of the Diagnosing stages, while partially influencing the Initiating and Establishing stages. Figure 3 illustrates the areas of IDEAL that the Agile Adoption Framework addresses.

The Agile Adoption Framework addresses the objectives of the Initiating phase of IDEAL via the first stage of its 4-stage process, Identifying Discontinuing Factors. While this first stage does not map exactly to the activities highlighted in the original IDEAL model, this stage assesses the organization to determine whether any factors are present that would prevent the adoption process from continuing. As in the initiating phase of IDEAL, one of factors to be assessed in the first stage of the 4-stage process is sponsorship support for the process improvement initiative.

In place of the Diagnosing phase of IDEAL, the Agile Adoption Framework uses the 4-Stage process and the agile measurement index to determine a target agile level for a project and develop a set of recommended agile practices for the organization to adopt. Like SCAMPI, the Agile Adoption Framework also identifies the strengths or weaknesses in the organization relative to the practices recommended for adoption. However, unlike SCAMPI, the framework does not include a set of best practices that highlight how to overcome the weaknesses. Instead, the Agile Adoption Framework provides guidance on prioritizing the weaknesses to be addressed. This is how the Agile Adoption Framework influences the third stage of IDEAL, Establishing. The Acting and Learning stages of IDEAL are outside the scope of this research.

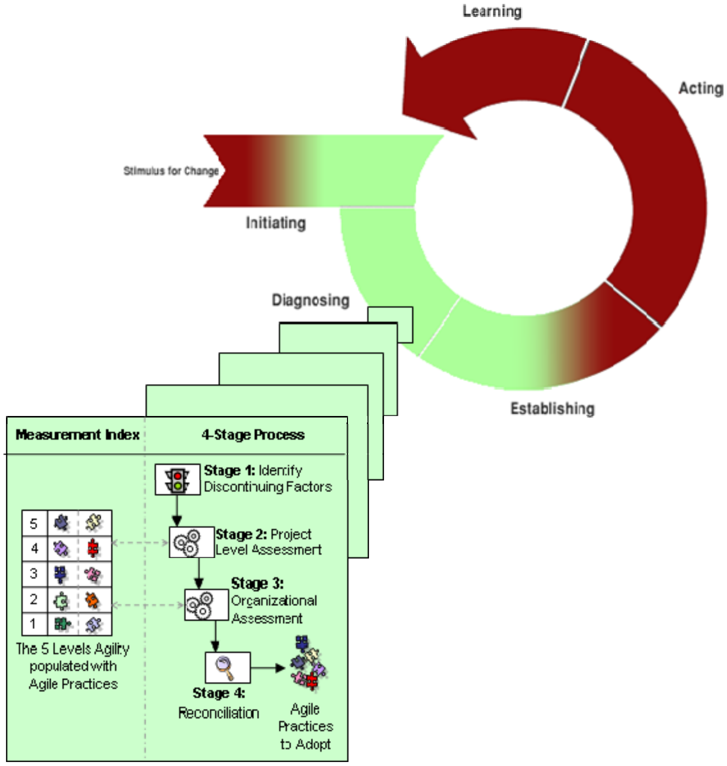

Learning
Acting
Stimulus for Change
Initiating
Diagnosing
Establishing
Measurement Index
4-Stage Process
Stage 1: Identify Discontinuing Factors
Stage 2: Project Level Assessment
Stage 3: Organizational Assessment
Stage 4: Reconciliation
Agile Practices to Adopt
5
4
3
2
1
The 5 Levels Agility populated with Agile Practices

Figure 3. Agile Adoption Framework relative to IDEAL

Chapter 3 provides a detailed discussion of the main component of the Agile Adoption Framework, the 4-Stage process. The chapter includes an explanation of how exactly each of the 4-stages helps provide guidance to organizations adopting agile practices from both a project and organizational perspective. Additionally, the chapter provides a brief overview of the Sidky Agile Measurement Index (SAMI). Chapter 4 then presents the details of the SAMI and an explanation of how it was created.

3. The Agile Adoption Framework: The 4-Stage Process

The Agile Adoption Framework is a structured and repeatable approach designed to guide and assist agile adoption efforts. It assists the agile community in supporting the growing demand from organizations that want to adopt agile practices. The main component of the Agile Adoption Framework is the 4-Stage Process, which utilizes the Sidky Agile Measurement Index (SAMI) to help an organization adopt agile practices. The SAMI is a scale the agile coach uses to identify the agile potential of a project or organization.

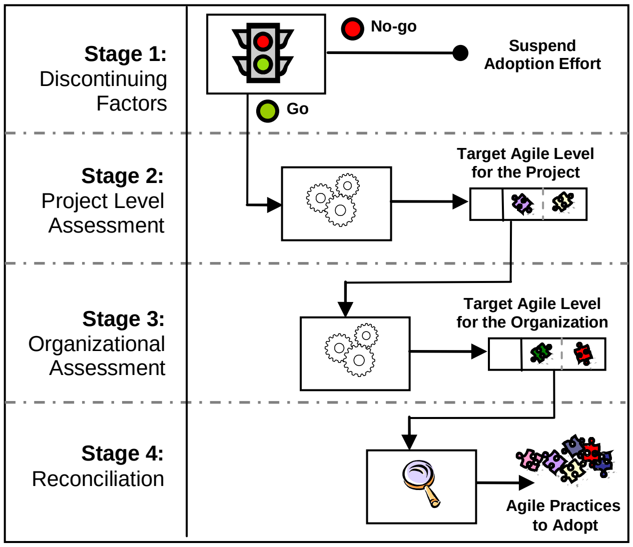


Figure 4. The 4-Stage Process for Agile Adoption

As depicted in Figure 4, the 4-Stage Process consists of four pieces that work together to help the assessor determine if (or when) an organization is ready to move toward agility, or, in other words, make the go/no-go decision, and assists him or her in the process of identifying which agile practices the organization should adopt. The four stages are:

- *Stage 1: Discontinuing Factors*. Discovers the presence of any roadblocks (or showstoppers) that can prevent the adoption process from succeeding

- *Stage 2: Project Level Assessment*. Utilizes the SAMI to determine the target level of agility for a particular project
- *Stage 3: Organizational Readiness Assessment*. Uses the SAMI to assess the extent to which the organization can achieve the target agility level identified for a project
- *Stage 4: Reconciliation*. Determines the final set of agile practices to be adopted by reconciling the target agile level for a project (from Stage 2) and the readiness of the embodying organization (from Stage 3)

This chapter focuses on the "backbone" of the Agile Adoption Framework, the 4-Stage Process. The chapter starts by presenting some of the challenges encountered in the development of the 4-Stage Process. Section 3.2 presents an overview of the SAMI, since the 4-Stage Process utilizes it. (Chapter 4 provides the details of the SAMI.) The latter part of this chapter includes sections on each of the four stages of the 4-Stage Process.

3.1. Motivation and Challenges

The objective behind the creation of the Agile Adoption Framework is to provide organizations with guidance on which agile practices to adopt. Similar to CMMI and SCAMPI, predecessors of the Agile Adoption Framework, the framework consists of two main components, an agile measurement index and a process component that uses the measurement index to provide the organization with guidance. The first three phases of IDEAL (Initiating, Diagnosing and Establishing) provide the primary sources of inspiration in determining the nature and structure of the process component of the framework. The objective was not to replicate these phases of IDEAL, but to learn from them what is needed to guide an organization through the adoption of a new technology, including agile practices.

Analysis of the objectives of these three stages of IDEAL results in a *process* that comprises the following four structured stages:

1. Discontinuing Factors
2. Project-Level Assessment
3. Organizational Readiness Assessment
4. Reconciliation

The decision to call this a process is based on the definition of the word *process*. According to the *American Heritage Dictionary*, one definition for a process is a series of actions, changes, or functions bringing about a result. Since these four stages do exactly this, they have been titled the *4-Stage Process*. The details of each of these four stages follow, but are proceeded by a discussion of two of the main issues that influenced the creation of these four stages. The first issue is related to the need of conducting pre-assessments, and the second concerns whether the desired state of agility should be addressed from a project-level perspective or an organizational level.

3.1.1 The Need to Conduct a Pre-Assessment

Traditional Software Process Improvement (SPI) models, based on the Capability Maturity Model (CMM) and CMM Integration (CMMI), recommend that the decision to start a SPI initiative be made after a trained assessor has conducted a pre-assessment of the organization in order to determine whether it is ready for SPI [43]. Organizations that do not embody the factors necessary for a successful SPI effort are considered “not ready.” In this situation, the SPI effort is suspended until the missing factors become present. Pre-assessment is also important in an agile adoption initiative, because it helps identify factors in an organization that can prevent the successful adoption of agile practices. If such factors exist, the organization must eliminate them before continuing with the adoption effort. This pre-assessment phase saves the organization time, money and effort by identifying upfront missing or existing factors that can cause the SPI or agile adoption effort to fail [53].

Additionally, conducting an assessment in order to decide whether or not to go ahead with the effort to introduce agile practices is important because of additional losses that *Technical Chaos* can precipitate. Technical chaos, the disruptions caused by the partial adoption of new practices, leaves the development process in an unstable state until it reverts back to the original engineering practices used before the failed adoption effort. Technical chaos is likely to occur when an agile adoption effort starts and then fails before completion. As a result, the stage titled Discontinuing Factors was created to ensure that the organization makes the decision to adopt agile practices only after finding no factors present that can jeopardize the adoption effort.

The next issue that influenced the development of the 4-Stage Process was whether the "desired state" of agility should be addressed from an organizational perspective or a project perspective.

3.1.2 Desired State: Project or Organization?

This challenge exists because most of the traditional SPI models and frameworks have an organizational focus. It is common in SPI to identify the current level of process maturity of the whole company (current state of the organization) and then determine the desired level for the whole organization (desired state of the organization). IDEAL, CMMI and the ISO 15504 discuss the current states or desired states in terms of the whole organization, and not in terms of current or desired states for individual projects. This focus makes sense, because these SPI initiatives are working to instill certain principles and Key Process Areas (KPAs) across the whole organization. Their objective is to see these principles and KPAs evident for every project in the organization.

However, adopting agile practices is different because the focus is not on principles or KPAs, but on the adoption of agile practices. The goal of agile adoption initiatives is to ensure the implementation of these principles and KPAs in the organization through the use of agile practices. For example since Requirements Management is a KPA for CMM Level 2, the objective is to ensure that agile practices are used to fulfill the goals of this KPA. This explains why adopting agile practices in an organization is not in opposition to achieving higher CMM levels. Moreover, within the same organization, the KPA of Requirements Management can be manifested through different practices, depending on the project. The requirements management needed for a government project differs in the way it is implemented from that needed for a social networking portal. Since every project is different and is surrounded by unique circumstances (customers, location, team and so forth) the agile practices used for one project are often unsuitable for another project.

The above discussion illustrates why traditional SPI initiatives address the "desired state" issue from an organizational perspective and why SPI initiatives that focus on adopting agile practices need to address the same issue, but from a project perspective. This conclusion answers the initial challenge of whether to address the "desired state" of agility from a project perspective or from an organizational perspective. The answer is that both need to be addressed. Since each project is different, the agile practices that are suitable for one project may or may not be suitable for another. Therefore, each project needs a specific *target level* of agility (the desired state). At the same time, it is important to recognize that this project is "living" within an organization and, therefore, it is necessary to take into account the organization too. The organizational assessment determines if the *organization is ready* for the project to adopt its target agile level.

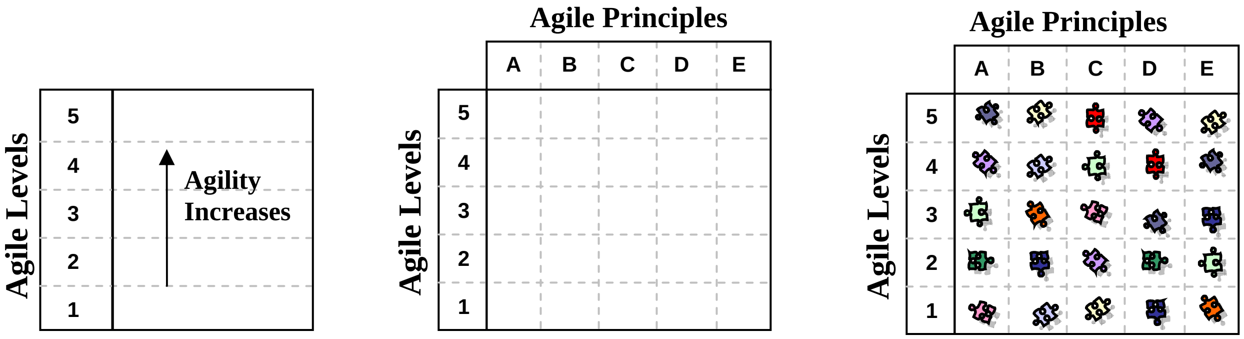
Since there was no stage responsible for developing recommendations, a fourth stage titled *Reconciliation* has been added to the Agile Adoption Framework. The Reconciliation stage is responsible for comparing the target project agile level and the organization's readiness level, and reconciling the differences between the two levels in a manner that yields recommendations on how to reach the desired agile level for the project.

The 4-Stage Process is assists organizations in determining their readiness to undertake an initiative to adopt agile practices. Additionally, the four stages help organizations determine which practices to adopt while identifying organizational weaknesses that might affect their adoption. After completing the 4-Stage Process, an organization needs to create a plan to strengthen the identified weaknesses, and plan for the adoption of the identified agile practices. With these plans in place, the organization adopts the practices and then confirms the success of the adoption. As mentioned in Chapter 2, the steps to complete agile adoption (occurring after the completion of the four stages of the framework) are outside the scope of this research.

The remainder of this chapter provides a detailed explanation of each of the four stages, including how each contributes to the overall objective of providing guidance to organizations aspiring to adopt agile practices. However, since the 4-Stage Process relies on the Sidky Agile Measurement Index (SAMI) for many of its assessments, the next section provides a brief description of that measurement index. Chapter 4 presents a detailed description of the SAMI.

3.2. Overview of the Sidky Agile Measurement Index (SAMI)

One of the main concerns organizations have when seeking to adopt agile practices is determining how agile they can become [39]. The agile potential (i.e. the degree to which that entity can adopt agile practices) of projects and organizations is influenced by the circumstances surrounding them. To determine the agile potential the coach (or the one conducting the assessment) needs to use a measurement index or scale that can assess the potential agility of an entity. The Agile Adoption Framework uses the Sidky Agile Measurement Index (SAMI) to determine the agile potential of a project or organization. The SAMI that is composed of four components:



(a) Empty Agile Levels

(b) Empty Agile Levels with Agile Principles

(c) Agile Levels populated with Agile Practices categorized within Agile Principles

Figure 5. Components of the Agile Measurement Index (Indicators are not shown)

1. **Agile Levels:** a set of agile practices that are related and, when adopted collectively, make significant improvements in the software development process, thereby leading to the realization of a core value of agility
2. **Agile Principles:** guidelines that need to be employed to ensure that the development process is agile
3. **Agile Practices and Concepts:** the concrete activities and practical techniques used to develop and manage software projects in a manner consistent with the agile principles
4. **Indicators:** questions the examiner uses to assess certain characteristics of an organization or project, such as its people, culture and environment, in order to ascertain the readiness of the organization or project to adopt an agile practice.

Chapter 4 details the development process of the SAMI and each of its components. This section presents a simple overview of the measurement index as a whole, without focusing on any particular component, as well as the name and objective of each of the five levels. Figure 5 shows the components of the SAMI.

Agile levels, as depicted in Figure 5a, are considered the units of the SAMI as they enumerate the possible degrees of agility for a project or organization. The agile potential of a project or organization is expressed in terms the highest agile level it can achieve. The attainment of a particular level indicates that the project or organization has realized and embraced the essential element needed to establish a commensurate agile development process. For example, when the elements inherent to *enhancing communication and collaboration* are embodied within the development process, then the Agile Level 1 (*Collaborative*) is attained. However, before a project can expect to move to Level 2 status, all practices associated with Agile Level 1 must be achieved or achievable. The agile levels represent the core qualities of agility as defined by the Agile Manifesto [2]. The objective of the level refers to the agile quality the level seeks to achieve or introduce into the development lifecycle. Table 2 shows the five levels in descending order.

Agile Level	**Level Name**	**Level's Objective (Agile Value Re-worded)**
Level 5	Encompassing	Establishing a vibrant and all-encompassing environment to sustain agility
Level 4	Adaptive	Responding to change through multiple levels of feedback
Level 3	Effective	Developing high quality, working software in an efficient an effective manner
Level 2	Evolutionary	Delivering software early and continuously
Level 1	Collaborative	Enhancing communication and collaboration

Table 2. The 5 Agile Levels in decsending order

Each of the agile levels is composed of a set of agile practices, that when adopted, helps accomplish the level's objective. The second component of the measurement index, *agile principles*, guides the assignment of agile practices and concepts assigned to each level.

Agile principles are the essential characteristics that must be reflected in a process before it is considered *Agile*. For example, two key agile principles are *human-centric*, which refers to the reliance on people and the interaction between them, and *technical excellence*, which implies the use of procedures that produce and maintain the highest quality of code possible. The SAMI uses five agile principles to ensure that the agile levels embody the essential characteristics of agility. Figure 5b illustrates the relationship between agile levels and agile principles. The top row of Table 3 enumerates those principles.

Each agile level contains practices associated with most, if not all, of the agile principles. The principle reflects the approach that the agile practice uses to promote the agile quality pertinent to a level. For example, all of the practices in Level 3 (Effective) promote the agile objective of *developing high quality, working software in an efficient an effective manner.* How an objective is achieved is determined by the practices associated with agile principles spanning each level. Along the same lines, practices associated with the *technical excellence* principle promote its agile objective by focusing on enhancing the technical aspect of the process; while practices associated with the *human-centric* principle promote enhancing the human aspect of the process. More about agile principles can be found in Chapter 4.

The real essence of the agile measurement index, however, is the agile practices it enunciates. Agile practices are concrete activities and practical techniques used to develop and manage software projects in a manner consistent with the agile principles. For example, *paired programming*, *user stories,* and *collaborative planning* are all agile practices. Since the agile levels are composed of agile practices (organized along the line of agile principles – see Figure 5c), they are considered the basic building block of the SAMI. The attainment of an agile level is achieved only when the agile practices associated with it are adopted. The SAMI is populated with 40 distinct agile practices. Table 2 illustrates these practices, arranged along the lines of the agile levels and principles.

A set of *indicators*, or *questions*, must accompany each agile practice or concept in the measurement index. The SAMI contains approximately 300 different indicators. The agile coach uses these indicators or questions to measure the extent to which the organization is ready to adopt an agile practice or concept. Each indicator is designed to measure a particular organizational characteristic necessary for the successful adoption of the agile practice the indicator is related to. Depending on the question, a manager, developer, or agile coach is designated to answer it, either subjectively or objectively. All the organizational characteristics that need to be assessed, along with the indicators needed to assess them, are placed in the Readiness Assessment Table (RAT) associated with each agile practice. RATs are explained in further detail in the discussion of the Indicators component of the SAMI in Chapter 4.

As an example of the above, assume the assessor wants to determine the extent to which the organization is ready to adopt *coding standards* (Level 1, Technical Excellence). The two organizational characteristics that need to be assessed are: (1) to what extent do the developers understand the benefits behind coding standards, and (2) how willing are they to conform to coding standards. Several indicators are used to assess each of these characteristics. For example, to assess willingness, the assessor might ask the developers to what extent would they abide by coding standards even when under a time constraint.

	Agile Principles				
	Embrace Change to Deliver Customer Value	*Plan and Deliver Software Frequently*	*Human-centric*	*Technical Excellence*	*Customer Collaboration*
Level 5 **Encompassing** *Establishing a vibrant environment to sustain agility*	Low process ceremony [60, 72]	Agile project estimation [29]	Ideal agile physical setup [60]	Test driven development [16] Paired programming [84] No/minimal number of level -1 or 1b people on team [24, 22]	Frequent face-to-face interaction between developers & users (collocated) [17]
Level 4 **Adaptive** *Responding to change through multiple levels of feedback*	Client driven iterations [60] Continuous customer satisfaction feedback [64, 77]	Smaller and more frequent releases (*4-8 weeks*) [64] Adaptive planning [60] [29]		Daily progress tracking meetings [8] Agile documentation [73, 57] User stories [30]	Customer immediately accessible [22] Customer contract revolves around commitment of collaboration [45, 64]
Level 3: **Effective** *Developing high quality, working software in an efficient an effective manner*		Risk driven iterations [60] Plan features not tasks. [29] Maintain a list of all features and their status (backlog) [57]	Self organizing teams [60, 72, 57, 27] Frequent face-to-face communication [72, 27, 18]	Continuous integration [60] Continuous improvement (refactoring) [57, 17, 41, 7]. Unit tests [50] 30% of level 2 and level 3 people [24, 22]	
Level 2: **Evolutionary** *Delivering software early and continuously*	Evolutionary requirements [60]	Continuous delivery [60, 57, 45, 17] Planning at different levels [29]		Software configuration management [57] Tracking iteration progress [60] No big design up front (BDUF) [5, 17]	Customer contract reflective of evolutionary development [45, 64]
Level 1: **Collaborative** *Enhancing communication and collaboration*	Reflect and tune process [64, 77]	Collaborative planning [72, 27, 60]	Collaborative teams [80] Empowered and motivated teams [18]	Coding standards [51, 82, 68] Knowledge sharing tools [60] Task volunteering [60]	Customer commitment to work with developing team [18]

Table 3. The Sidky Agile Measurement Index (SAMI) populated with agile practices and concepts

The SAMI is used in the process component of the framework, which consists of four stages working together to guide organizations in identifying agile practices that best fit into their environment. After this brief introduction to the SAMI, the remainder of this chapter illustrates how the 4-Stage Process uses the SAMI to provide organizations with guidance and assistance.

3.3. Stage 1: Discontinuing Factors

The objective of the first stage of the 4-Stage Process is to provide organizations with a method for reaching a decision on whether or not to proceed with agile adoption initiatives. As Figure 6 shows, to achieve this goal, Stage 1 provides organizations with an assessment process that identifies the factors that could prevent the successful adoption of agile practices. These are called *discontinuing factors* and can vary from one organization to another. Stage 1 suggests that organizations follow three steps in order to fulfill the objective of this stage:

1. Determine the list of discontinuing factors
2. Assess the extent of the presence of discontinuing factors
3. Make Go/No-go decision based on assessment results

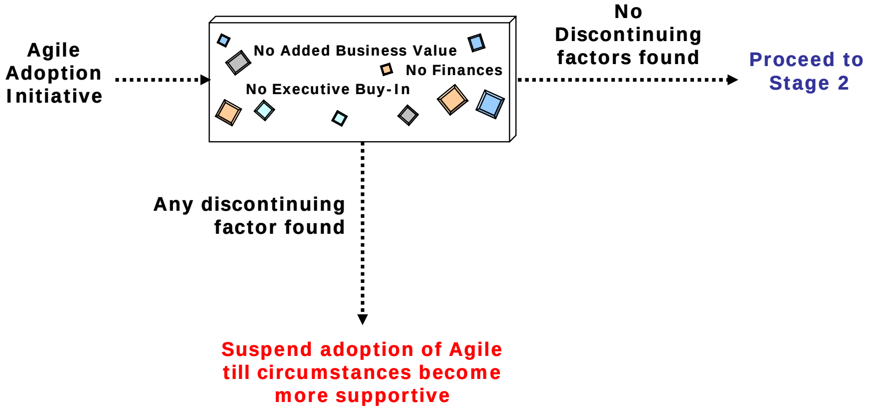

Agile Adoption Initiative
No Added Business Value
No Finances
No Executive Buy-In
No Discontinuing factors found
Proceed to Stage 2
Any discontinuing factor found
Suspend adoption of Agile till circumstances become more supportive

Figure 6. Stage 1: Discontinuing Factors

The following subsections provide a discussion of the importance of this stage and detailed descriptions of the actual steps that take place during this stage.

This discussion begins with a description of how Stage 1 of the process guides and assists organizations in making Go/No-go decisions concerning the adoption of agile practices. A pre-assessment activity that identifies any discontinuing factors aids in making this decision. The next three sections provide a detailed discussion of the process that should be followed during this stage.

3.3.1 Determining the Discontinuing Factors

The first step in Stage 1 of the 4-Stage Process is to identify the factors that could adversely impact the agile adoption process. These *Discontinuing Factors* are organizational characteristics that, if present in an organization, can hinder or jeopardize the success of the agile adoption process. These factors can vary from organization to organization and from one agile consultant to another. They typically pertain to an organization's resources including money, time and effort, as well as the support of the executives.

The IDEAL model is a source of inspiration for identifying discontinuing factors. The Initiating phase of IDEAL helped identify two factors that, if absent from an organization, could prevent the success of the agile adoption effort. These two discontinuing factors are:

- *Inappropriate Need for Agility:* This refers to situations where, from a business or software development perspective, adopting agility does not add any value. This factor was derived from the initial input of the initiating phase of IDEAL, Stimulus for change.
- *Absence of Executive Support:* If committed support from executive sponsors is absent, then effective and substantial change in the organization is unlikely

to occur. This factor was also derived from the initiating phase of IDEAL that emphasizes the presence of Sponsorship before the start of the SPI effort.

A review of various adoption cases validates the identification of these two factors as discontinuing factors. Several authors note that these two factors could hinder the adoption process if present. For example, the first two factors that Spayd [79] highlights as needed to realize success in the adoption of agile practices are the counterparts of both these factors derived from IDEAL. Cohn [31], Eckman [37], and Pukinskis [71] all support the idea that the lack of executive buy-in or sponsorship is a factor that can jeopardize the success of any process improvement effort, especially relative to agile.

A third discontinuing factor is the *lack of sufficient funds*. When funds are unavailable or insufficient to support the agile adoption effort, then an adoption process is not feasible [37]. As obvious as this factor is, it is important to be conscious of it, especially if the change effort is going to be managed in-house. Usually, if a consultant is hired to conduct the adoption process, he or she makes sure that sufficient funds are available before the engagement starts.

Another discontinuing factor initially included is the *type of the project*, because mission or life-critical projects are not suitable candidates for agility. This assumption is based on the work of Boehm and others who argue agile development is not suitable for mission and life critical systems [4] [69] [19] [24]. However, since more and more agile development can be found in the mission and life-critical projects [56] [54] [35], some of the authors that maintained this position are now changing their point of view. Therefore, the type of project no longer needs to be a discontinuing factor, even though the level or degree of agility used for mission and life-critical systems might be different. This realization resulted in a paper showing how to use the concepts of the agile adoption framework to identify the practices suitable for the development of mission and life-critical systems [78].

As mentioned earlier, these are not the only discontinuing factors. Other organizations or consultants can identify other discontinuing factors. However, the key to identifying these factors is to think of what could stop or hinder the agile adoption process, regardless of the number of agile practices being adopted.

When an organization demonstrates *any* of these discontinuing factors, it is unprepared to move towards agility and should suspend the adoption process until the environment is more supportive. With the discontinuing factors identified, the next step is to employ the process used to assess their presence in the organization.

3.3.2 Assess the Presence of Discontinuing Factors

Once the assessor or the organization has identified the discontinuing factors, the next step toward agile adoption is to ascertain the extent of the presence or absence of these factors. Like the approach used by the SAMI, this step relies on indicators to assess the degree to which a discontinuing factor is present or absent. Indicators are questions that people in the organization or the assessor himself or herself answer. Depending on the factor, indictors measure the specific organizational characteristics that directly influence the existence of that discontinuing factor.

For example, to determine whether an organization lacks the sufficient funds for the agile adoption effort, one of the organizational characteristics measured is the dollar amount of funds allocated to the process improvement effort. Another characteristic measured is the ability to actually spend the funds for agile adoption. At least one indicator, though more are preferable, is used to assess each of these characteristics. An example of a question or indicator used to assess the ability to spend funds on agile adoption is *Can the funds be spent on any process improvement activity?* Another indicator is *Are there any restrictions on the type of activities these funds can be used for?*

Each discontinuing factor has an assessment table that highlights the different organizational characteristics that the assessor needs to assess in order to determine the extent to which the factor is present.

Category of Assessment	Area to be assessed	Characteristic(s) to be assessed	To determine:	Assessment Method	Sample Indicators
Project	Requirements	Rate of Change	Whether or not the project's requirements are clear and well defined, thus predicting no change, or whether or not the requirements need to be flexible and/or might change	Interviewing	DC_M5, DC_M6, DC_M7
	Delivery	Time to Market	Whether or not the project has to be developed quickly in order to introduce it to the market as soon as possible	Interviewing	DC_M4
Organization	Project History	Schedule and Budget	Whether or not the organization has a trend of having projects that go over time and budget	Observation	DC_A1, DC_A2
	Software Process	Problems	Whether or not the organization is facing any problems or dissatisfaction with the current software process	Interviewing	DC_D1, DC_D2, DC_D3, DC_M1, DC_M2, DC_M3

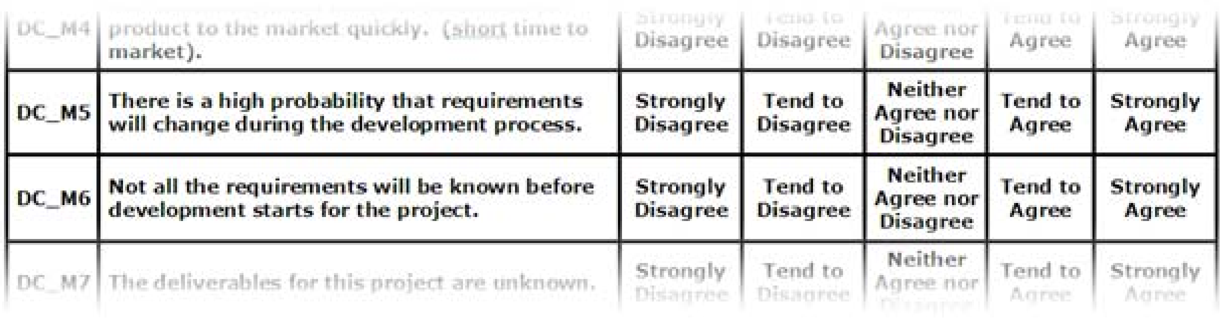
Table 4. Assessment Table for the Discontinuing Factor: Inappropriate Need for Agility

For example, Table 4 depicts the assessment table used for the first discontinuing factor, Inappropriate Need for Agility. The assessor uses this factor to determine whether or not there is any value added by adopting agile software development. The first vertical column in the table identifies the generic organizational area to be assessed. The next vertical column specifies which aspect of the organizational area needs assessment. The third column designates the actual organizational characteristic to be assessed. In this example, the assessor needs to assess four different characteristics in order to measure the existence or absence of this discontinuing factor. One of these characteristics is the *rate of change* of the project requirements. The fourth column, "To determine," defines the goal behind the assessment of this characteristic. The fifth column provides information on the method used to conduct the assessment and the last column provides a reference to the actual indicators the assessor uses for each particular organizational characteristic. The indicator number is used to identify and reference the indicator. The first two letters in the indicator's number (DC) mean that the assessors use these indicators for assessment of DisContinuing factors. The letter after the underscore (_) refers to the type of person who should provide an answer for the indicator:

- *A*: denotes an indicator that the assessor, or the person conducting the assessment, needs to answer

- *D*: denotes an indicator that the developer, or anyone on the development side of the project, needs to answer
- *M*: denotes an indicator that a manager, or anyone performing management related tasks for to the project, needs to answer

A sequential number is used as the last digit in the indicator's number. Figure 7 shows some sample indicators, which usually consist of a statement or question that needs a response. Most indicators are based on a 5-point Likert summated scale, from 1 "strongly disagree" to 5 "strongly agree." A small number of other indicators are based on other 5-point scales that are more appropriate to the organizational characteristic being assessed.



DC_M4	product to the market quickly. (short time to market).	Strongly Disagree	Tend to Disagree	Agree nor Disagree	Tend to Agree	Strongly Agree
DC_M5	**There is a high probability that requirements will change during the development process.**	**Strongly Disagree**	**Tend to Disagree**	**Neither Agree nor Disagree**	**Tend to Agree**	**Strongly Agree**
DC_M6	**Not all the requirements will be known before development starts for the project.**	**Strongly Disagree**	**Tend to Disagree**	**Neither Agree nor Disagree**	**Tend to Agree**	**Strongly Agree**
DC_M7	The deliverables for this project are unknown.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree

Figure 7. Sample Indicators for Discontinuing Factors

Figure 8 shows the organizational characteristics that need to be assessed for each of the discontinuing factors. The assessor uses 21 different indicators to assess the discontinuing factors. Appendix A contains the assessment tables for all three factors along with all 21 indicators used to assess them.

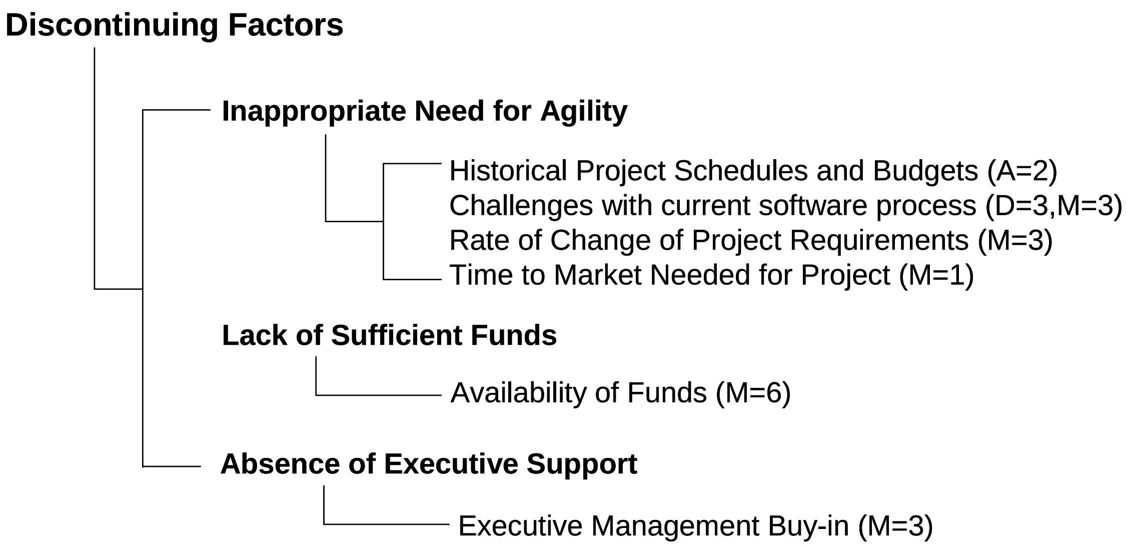

Discontinuing Factors
Inappropriate Need for Agility
Historical Project Schedules and Budgets (A=2)
Challenges with current software process (D=3,M=3)
Rate of Change of Project Requirements (M=3)
Time to Market Needed for Project (M=1)
Lack of Sufficient Funds
Availability of Funds (M=6)
Absence of Executive Support
Executive Management Buy-in (M=3)

Figure 8. Hierarchy of Indicators for Discontinuing Factors

Appendix A also highlights the method used to aggregate the results of the different indicators to come up with one nominal value, as shown in Table 5. The aggregate result of the answers to the indicators (the nominal value) shows the extent to which that factor is present or absent in the organization. The decision to proceed with the adoption effort, or abandon it, is based on the nominal values for all the factors. The next section provides a discussion of this step.

Not Achieved	**0%-35%**
Partially Achieved	**35%-65%**
Largely Achieved	**65%-85%**
Fully Achieved	**85% - 100%**

Table 5. Nominal Values for Aggregated Indicators

3.3.3 Make Go/No-go Decision

Once the assessment process has identified the degree to which each discontinuing factor is present in the organization, it moves to the third step within Stage 1 of the 4-Stage Process of the Agile Adoption Framework. During this step, the assessor recommends proceeding with the agile adoption effort or abandoning it based on the absence of discontinuing factors. To receive the “green light,” the degree of *absence* for each of the discontinuing factors must fall below an acceptable threshold for the organization. While this seems a little counterintuitive at first, since the assessment is for discontinuing factors whose presence is a hindrance to the adoption process, their absence is a good sign. In other words, the goal is to have all the discontinuing factors score a “Not Achieved.” The greater the presence of discontinuing factors in the organization, the worse the situation is. For example, assume Table 6 shows the assessment results for an organization. The organization has decided to go ahead with the agile adoption effort when the level of discontinuing factors was found to be below 35% (Not Achieved). However, because one or more of the discontinuing factors was found at a level higher than the threshold, the assessor has recommended a *No-go* decision. Since the level of presence of the discontinuing factor, *Lack of Sufficient Funds,* was higher than 35%, there is a No-go decision to continue the agile adoption effort. The adoption process is suspended until the conditions changed and a reassessment showed that the presence of the discontinuing factor had dropped below the threshold.

Discontinuing Factors	**Not Achieved**	**Partially Achieved**	**Largely Achieved**	**Fully Achieved**
	0%-35%	**35%-65%**	**65%-85%**	**85%-100%**
Inappropriate Need for Agility	X			
Lack of Sufficient Funds		X		
Absence of Executive Support	X			

Table 6. Assessment Results for Discontinuing Factors

Each organization can set its own thresholds for different factors. For example, one organization might decide to suspend the adoption process if any discontinuing is partially achieved or higher (35% to 100%). However, another organization might have the same rule, but set a higher threshold for one of the factors. For example, the organization might tolerate a higher threshold (below 65%) for only the factor related to funds. In this case, the example shown in Table 6 would generate a *Go* decision because the funds factor is below its threshold (65%) and the other factors are already “Not Achieved”. Again, these thresholds are left to the stakeholders to determine.

Clearly, it is necessary to ensure that organizations have the capability to introduce agile practices into their process before starting such a process improvement effort. If they are not ready, then they may unnecessarily commit to an initiative that can have detrimental consequences later.

In summary, Stage 1 provides guidance to organizations needing to decide whether to start the agile adoption effort. Identifying and assessing the absence of discontinuing factors in the organization determines an organization’s ability to proceed with the introduction of agile practices. Once the stakeholders decide to proceed with the agile adoption effort, the guidance process enters the second stage of the 4-Stage Process – the project-level assessment.

3.4. Stage 2: Project-Level Assessment

After the stakeholders make the decision to go ahead with the agile adoption effort (from Stage 1), the next stage looks at the individual projects that will adopt agile practices and determines which level of agility (based on the SAMI) each should adopt. Since each project is different and is surrounded by unique circumstances, each project needs to adopt a level of agility that is best suitable for it.

Figure 9 depicts an outline of the approach used in Stage 2 to determine the target agile level for each project. The assessor starts by assessing certain factors related to Agile Level 1. If this assessment is positive, the assessor proceeds to assess factors related to the next agile level. The level at which the assessment fails is identified as the highest agile level the project can reach (i.e. target agile level).

Section 3.4.2 describes the details of how limiting factors outside the organization’s control, influence the target level established for each project. These limiting factors are discussed from an organizational perspective as well as a practice-based perspective. But first, Section 3.4.1 gives some brief background about the notion of project-based agility.

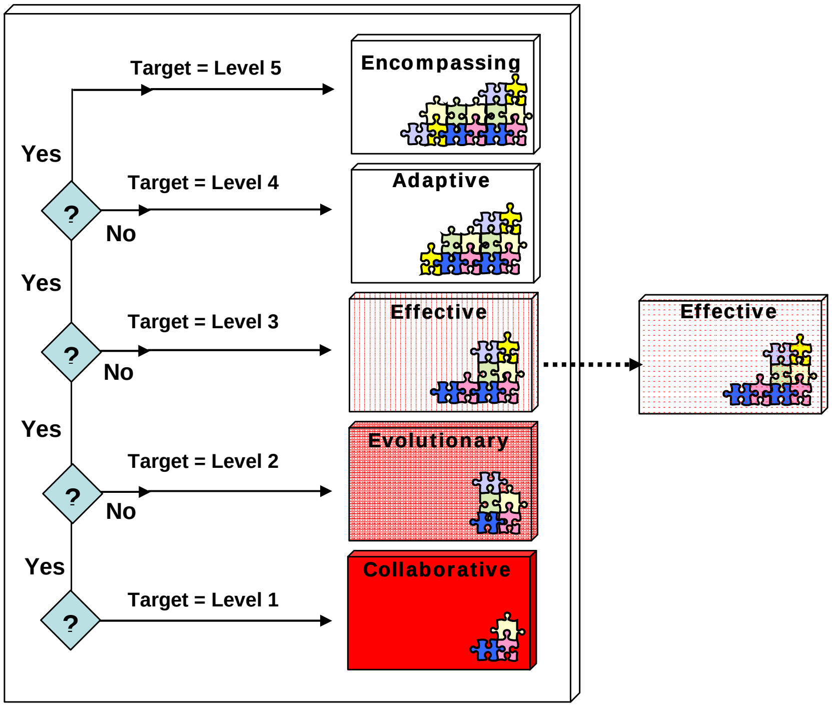


Figure 9. Stage 2 : Project Level Assessment

3.4.1 Project-Level Assessment: Background Information

Practitioners and researchers alike accept the idea of having different degrees of agile software development based on the type of project. One of the more famous agile methodologies based on the premises that each project is different and, therefore, necessitates different degrees of agility (or process) is Cockburn's Crystal Family [24] [26]. The Crystal family includes a number of different methodologies and the criticality and size of each individual project guides the choice of the most suitable methodology for the project.

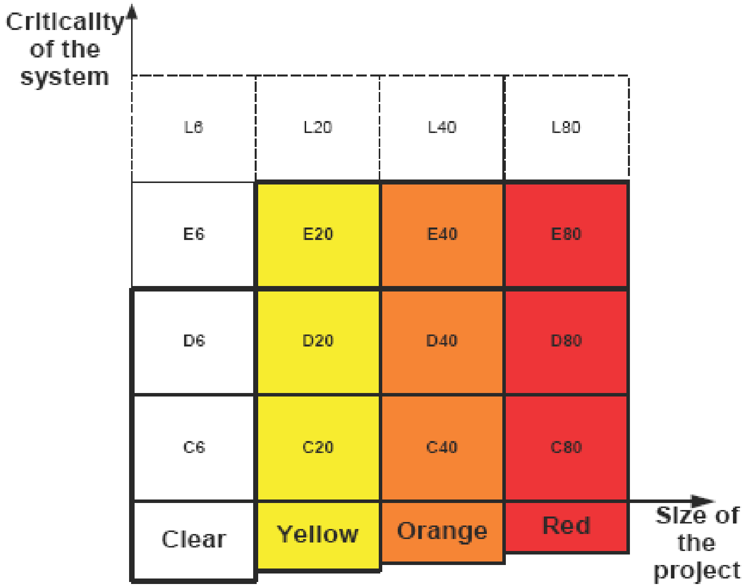


Figure 10. Crystal Family of Methodologies [24] [26]

Figure 10 shows that each member of the Crystal family is marked with the color indicating the heaviness of the methodology, i.e. the darker the color the heavier the methodology. The character symbols indicate the potential loss caused by a system failure (i.e. C=Comfort, D=Discretionary Money, E=Essential Money, and L=Life). The numeral beside the character symbol represents the number of the people on the project. The importance of the Crystal family is that Cockburn gives each project a different methodology depending on the criticality and size factors of the project.

Little followed the same concept as Cockburn with his Adoptive Agility but identified factors other than criticality and team size to determine the appropriate methodology for the project. Little classified projects according to Complexity Attributes (which included team size and criticality) and Uncertainty Attributes [63] [62]. Based on the assessment of these two sets of attributes, a project is classified into one of four project types:

- Dogs – Simple projects with low uncertainty
- Colts – Simple projects with high uncertainty

- Cows – Complex projects with low uncertainty
- Bulls – Complex projects with high uncertainty

Little did not define a set methodology for each of the four project types, but he highlighted some agile practices that are suitable for each project type.

Both Cockburn's Crystal Family and Little's Adaptive Agility demonstrate the possibility of selecting different factors to determine the right degree of process or process agility for a particular project. The development of Stage 2 of the 4-Stage Process is indebted to the work of Cockburn and Little. However, Stage 2 differs from the work of others, because the level of agility a project can reach is not based on either criticality and size as the Crystal Family or the complexity and uncertainty, as Adaptive Agility is, but on the absence of factors outside the organization's control that are needed to adopt an agile practice. The next sections present, in greater detail, the approach Stage 2 uses to determine the target level of agility for a project.

3.4.2 Establishing a Target Agile Level

Stage 2 is the first stage of the adoption process that utilizes the SAMI presented briefly in Section 3.2. The objective of this stage is to identify the highest level of agility a project can achieve. This is called the target agile level and is one of the five agile levels defined in the SAMI.

The important question is how the target level is identified for each project. The first step to answering this question is to recall the design of the SAMI. In it, each agile level is composed of a set of agile practices. Each one of these practices is associated with a set of indicators that assesses different aspects or factors of the organization to determine the extent the organization is ready to adopt that agile practice. The organization cannot change some of the organizational factors or aspects assessed by these indicators, because they exist outside of the organization's control. These outside factors influence the target agile level for each project.

For example, *frequent face-to-face communication* is an agile practice at Level 3. Therefore, *near team proximity* (having the whole team in a close proximity to each other) is a factor needed to successfully adopt this practice. If the project and organization have no say in changing this project characteristic (*near team proximity)*, because it is outside of their control, and if the project level assessment determines that this factor is not achieved for this project and can not be achieved, because it is outside the organization's control, then the highest level of agility for this project will be the same Agile Level in which this agile practice is found, which is Level 3 in this case.

For terminology purposes, organizational or project factors that the organization or project team cannot change are called *limiting factors* and the agile practices or concepts that depend on these factors for a successful adoption are called *limiting agile practices*. The name *limiting factor* was chosen, because each has the potential to limit the target agile level for a project.

Since identifying the target agile level, or the highest level a project can aspire to, depends on limiting factors, or factors outside of an organization's control, the first step is to identify all these limiting factors and any agile practices that depend on them for a successful adoption (limiting agile practices).

3.4.3 Limiting Factors: Organizational Perspective

There are many organizational and project characteristics that are assessed to determine whether the organization is ready to adopt an agile practice or not. The organization can change some of these factors, but not others. Different organizations have different factors that they have control over and others that they have no control over. The ability to change characteristics depends on the structure of the organization and its policies and procedures. Therefore, since the limiting factors vary from organization to organization, these factors exemplify a few of the more common possibilities:

- The Customer
- Team proximity
- Team competency

If an organization can change any of these factors, then, by definition, they no longer qualify as limiting factors. That said, the organization usually has little or no control over customer-related factors and, therefore, cannot expect to change them. Team proximity, however, is another matter. While it is something an organization can control, sometimes an organization fails to consider this factor when allocating a certain team for the project and finds itself hamstrung, because this allocation cannot be changed. For example, consider the situation where five of the 12-member team are half way across the country or the globe, and there is nothing that the project or organization can do (practically) to change this situation. However, if the organization can change the team members proximity, then by definition team proximity is no longer a limiting factor. The same logic applies to team competency. If the team allocated for a certain project contains no experienced people, and this is a factor the organization cannot change, then team competency is a limiting factor.

Therefore, at the start of each project or when working with an organization, it is important to know exactly which aspects or factors the organization can change and which it cannot change. This knowledge gives the person leading the effort to adopt agility a clear perspective of what to expect. Stage 2 does this, but in a structured and more detailed manner.

3.4.4 Limiting Factors: Practice Perspective

Once the person leading the adoption effort has identified the limiting factors, the next step is to identify any agile practices that depend on any of these limiting factors for a successful adoption. The Readiness Assessment Table (RAT) (explained in detail in Section 4.5.2) associated with each agile practice provides guidance for this step. These RATs identify each organizational characteristic that needs to be assessed. If any of the limiting factors are found in the RAT of any agile practice, that agile practice is identified as a Limiting Agile Practice. Based on the examples of limiting factors identified earlier in the Agile Adoption Framework, the limiting practices are:

Customer dependent:

- Frequent face-to-face interaction between developers and users
- Immediate customer accessibility
- Collaboration commitment in customer contract
- Evolutionary development reflected in customer contract
- Customer commitment to work with developing team

Team-proximity dependent:

- Frequent face-to-face communication
- Ideal agile physical setup

Team competence dependent:

- 30% of Cockburn's Level 2 (experienced) and Cockburn's Level 3 (highly experienced people
- No/minimal number of Cockburn Level 1 (no experience) or Cockburn Level 1b (some experience) people on team

Table 7 shows the position of all the potential limiting agile practices within the SAMI. It is important to note that the limiting agile practices identified in this section can vary depending on two things:

1. the limiting factors identified in the previous subsection
2. the set of readiness indicators associated with each agile practice.

	Agile Principles				
	Embrace Change to Deliver Customer Value	*Plan and Deliver Software Frequently*	*Human-centric*	*Technical Excellence*	*Customer Collaboration*
Level 5 **Encompassing** *Establishing a vibrant environment to sustain agility*	Low process ceremony	Agile project estimation	Ideal agile physical setup	Test driven development Paired programming No/minimal number of level -1 or 1b people on team	Frequent face-to-face interaction between developers & users (collocated)
Level 4 **Adaptive** *Responding to change through multiple levels of feedback*	Client driven iterations Continuous customer satisfaction feedback	Smaller and more frequent releases (*4-8 weeks*) Adaptive planning		Daily progress tracking meetings Agile documentation User stories	Customer immediately accessible Customer contract revolves around commitment of collaboration
Level 3: **Effective** *Developing high quality, working software in an efficient an effective manner*		Risk driven iterations Plan features not tasks. Maintain a list of all features and their status (backlog)	Self organizing teams Frequent face-to-face communication	Continuous integration Continuous improvement (refactoring) Unit tests 30% of level 2 and level 3 people	
Level 2: **Evolutionary** *Delivering software early and continuously*	Evolutionary requirements	Continuous delivery Planning at different levels		Software configuration management Tracking iteration progress No big design up front (BDUF)	Customer contract reflective of evolutionary development
Level 1: **Collaborative** *Enhancing communication and collaboration*	Reflect and tune process	Collaborative planning	Collaborative teams Empowered and motivated teams	Coding standards Knowledge sharing tools Task volunteering	Customer commitment to work with developing team

Table 7. The SAMI with Limited Agile Practices Underlined

For example, if an organization can change team competency, then all the practices that depended on this factor no longer become limiting agile practices. Moreover, if, for example, new readiness indicators that assessed the team proximity factor were added for self-organizing teams (an agile practice in Level 3 of the SAMI), then self-organizing teams become a limiting agile practice.

With the identification of potentially limiting agile practices, the assessor is ready to begin the project-assessment needed to identify the highest level of agility a project can reach (i.e. the target agile level). This process measures only the extent to which the factors needed for the limiting agile practices are present. The assessor conducts the assessment using the indicators associated with each agile practice. Since indicators are an essential component of the SAMI, Chapter 4 describes in detail how to use indicators to conduct assessments.

The assessor starts the process by assessing the organization's readiness for adopting only the limiting agile practices of the first agile level. If the result is positive and shows that the organization is ready to adopt those practices then the assessment process moves upward on the scale. The assessor does the same for the limiting practices identified in Agile Level 2 and so on. As long as the assessment results show that the organization *can* adopt the limiting practice, the assessment keeps moving to higher levels. Once the factors needed for the adoption of a limiting practice are absent, then the assessment process stops, and the highest level of agility for the project is the level at which the limiting practice is found.

The highest level of agility attainable is the level at which the assessment for the limiting factor has failed, because the factors needed for the adoption of a limiting agile practice are absent and the organization cannot do anything to change that. Therefore, the project cannot aspire to a higher level of agility, because it cannot fully adopt all the practices in the current level. If the factors outside the organization's control change, then the assessment process can continue to identify a new, and higher, target level for the project.

When factors outside the control of the organization constrain the highest level of agility for a project, the focus falls on resolving the constraining factors that prevent all the principles of agility from rising to higher levels of agility. Focusing on eliminating these factors is better and more beneficial than focusing on ways to adopt agile practices in higher levels because this does not resolve the effect the weakest principle exerts on the adoption of agility.

In summary, the target agile level for a project is the point when the assessment process discovers that one of the project characteristics needed to adopt a limiting agile practice or concept is missing, and neither the project nor organization can do anything to influence or change this. After the assessment has identified the target agile level, the next step in the journey to agility is to conduct an organizational readiness assessment to determine the set of agile practices that *can* be adopted for the project.

3.5. Stage 3: Organizational Readiness Assessment

Identifying the target level for a project does not necessarily mean that that level is *achievable.* Determining the achievable level requires an assessment of the readiness of the organization to adopt each of the agile practices up to, and including, the target level.

During Stage 3, the assessor conducts an organizational assessment to determine the extent of the readiness of the organization to adopt the agile practices contained in the target agile level. The activities of Stage 3 highlight the areas that need improvement to accommodate the adoption of the agile practices. In other words, the assessor uses this stage to develop a set of recommendations for achieving the target agile level for the project.

As Figure 11 shows, Stage 3 depends on the project's target level identified from Stage 2. Stage 3 assesses all of the different aspects of the organization to determine whether the organization has the necessary characteristics to support the practices encompassed in the target agile level. The details of how Stage 3 guides this organizational readiness assessment and the role the SAMI plays follow a brief discussion of the background of this stage and the importance of conducting a readiness assessment in the next section.

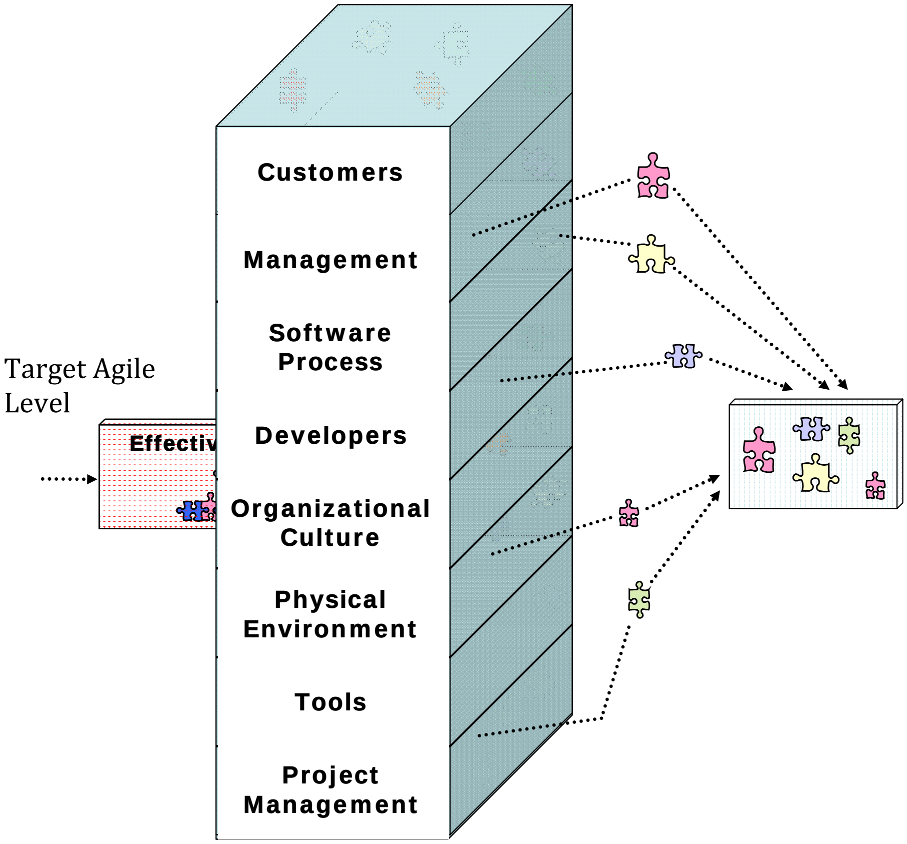


Figure 11. Stage 3: Organizational Readiness Assessment

3.5.1 Organizational Assessment : Background Information

The idea of conducting an assessment before the actual adoption process starts is not new. The first suggestion Boehm and Turner have for organizations trying to adopt agile practices is to incorporate preparation upfront [21]. They urge organizations to spend time and effort to conduct a significant upfront analysis to identify any mismatches between the organization and the set of agile practices it wants to adopt. Moreover, Ceschi *et al.* point out that one of the biggest challenges to introducing agile methods in an organization is the lack of a detailed preliminary evaluation of the challenges that this introduction might cause.

Although it is important to know whether the organization is ready to handle the adoption of certain agile practices *before* it starts adopting them, all too often the adoption efforts overlook this pre-adoption assessment phase or do not spend enough time and effort on it. The result is that an organization starts to adopt agile practices without knowing whether it is ready for them. Challenges start to emerge and hardships follow. The immediate solution in this kind of situation is either to try harder to adopt the practice or to abandon that the practice and deem it as an unsuitable practice. Stage 3 of the 4-Stage Process of the Agile Adoption Framework offers a solution to the readiness problem. As described in the next section, it provides the assessor with guidelines for assessing the readiness of the organization for each agile practice in such a way that it identifies, *before* any adoption effort starts, practices that are suitable and unsuitable for the *organization*. Moreover, the assessment process highlights the exact reasons when a practice is unsuitable for adoption.

Investing time and effort in this type of pre-adoption assessment of each agile practice increases the probability of success for the overall transition to agility [21]. Carrying out the activities within Stage 3 significantly reduces the risks associated with the agile adoption process. Furthermore, this assessment identifies the exact organizational characteristics that might jeopardize the successful adoption of an agile practice. Finally, all of this happens before the organization initiates the adoption effort and before the effort might cause disruption.

Knowing where an organization falls short before the adoption process helps it either select a more suitable set of agile practices to adopt or improve the weak organizational characteristics that can cause the adoption process to fail.

3.5.2 Accomplishing the Organizational Readiness Assessment

The main objective of Stage 3 is to determine the extent to which the organization is ready to achieve the project's target agile level. To do so, the assessor assesses how ready the organization is to adopt each of the agile practices within the target agile level, and the level(s) below it. The SAMI plays a crucial role during the assessment process of this stage.

Using the SAMI for Organization Readiness Assessment

Like Stage 2, Stage 3 of the 4-Stage Process also relies on the SAMI. However, Stage 2 assessed the readiness of agile practices that depended on organizational characteristics outside the organization's control. Stage 3 relies on the other set of agile practices within SAMI, those that depend on organizational characteristics that the project or organization *can change*. This is a fundamental difference in the use of the SAMI from Stage 2 to Stage 3.

Table 8 shows the SAMI. During Stage 2, the limiting agile practices (those not underlined) were assessed to determine the highest level of agility for a project. During Stage 3, the organization's readiness for the rest of the agile practices or the non-limiting Agile practices (those <u>underlined</u>) is assessed.

The indicators associated with each agile practice in the SAMI are instrumental in the assessment. They are used to determine the extent to which the organization is ready to adopt each individual agile practice. The indicators associated with each agile practice are concerned with assessing all the organizational characteristics that could influence the extent to which the organization is ready to adopt the agile practice. The type of organizational or project characteristics usually assessed to determine the readiness for a practice are:

- *Customers:* the project's customers and clients
- *Developers*: the technical staff involved with the development of the project

- *Managers*: the managers or executives overseeing the project and involved with decision making
- *Tools*: the software tools used within the organization or for a certain project
- *Culture*: the overall culture of the people within an organization or the project team

	Agile Principles				
	Embrace Change to Deliver Customer Value	*Plan and Deliver Software Frequently*	*Human-centric*	*Technical Excellence*	*Customer Collaboration*
Level 5 **Encompassing** *Establishing a vibrant environment to sustain agility*	Low process ceremony	Agile project estimation	Ideal agile physical setup	Test driven development Paired programming No/minimal number of level -1 or 1b people on team	Frequent face-to-face interaction between developers & users (collocated)
Level 4 **Adaptive** *Responding to change through multiple levels of feedback*	Client driven iterations Continuous customer satisfaction feedback	Smaller and more frequent releases (*4-8 weeks*) Adaptive planning		Daily progress tracking meetings Agile documentation User stories	Customer immediately accessible Customer contract revolves around commitment of collaboration
Level 3: **Effective** *Developing high quality, working software in an efficient an effective manner*		Risk driven iterations Plan features not tasks. Maintain a list of all features and their status (backlog)	Self organizing teams Frequent face-to-face communication	Continuous integration Continuous improvement (refactoring) Unit tests 30% of level 2 and level 3 people	
Level 2: **Evolutionary** *Delivering software early and continuously*	Evolutionary requirements	Continuous delivery Planning at different levels		Software configuration management Tracking iteration progress No big design up front (BDUF)	Customer contract reflective of evolutionary development
Level 1: **Collaborative** *Enhancing communication and collaboration*	Reflect and tune process	Collaborative planning	Collaborative teams Empowered and motivated teams	Coding standards Knowledge sharing tools Task volunteering	Customer commitment to work with developing team

Table 8. The SAMI with Non-Limited Agile Practices underlined

- *Project management*: the procedures and practices related to managing projects in the organization
- *Software process*: the activities and artifacts related to the software development process in the organization
- *Physical environment*: the physical layout of the organization and the geographical and geo-spatial distribution of its employees.

Figure 12, illustrates all the organizational characteristics that need to be assessed for each of the agile practices in Agile Level 1. Notice that the practice *Customer commitment to work with development team* is not part of the figure, despite the fact that it is one of the practices in level 1. This is again because Stage 3 deals only with the agile practices that are not limiting.

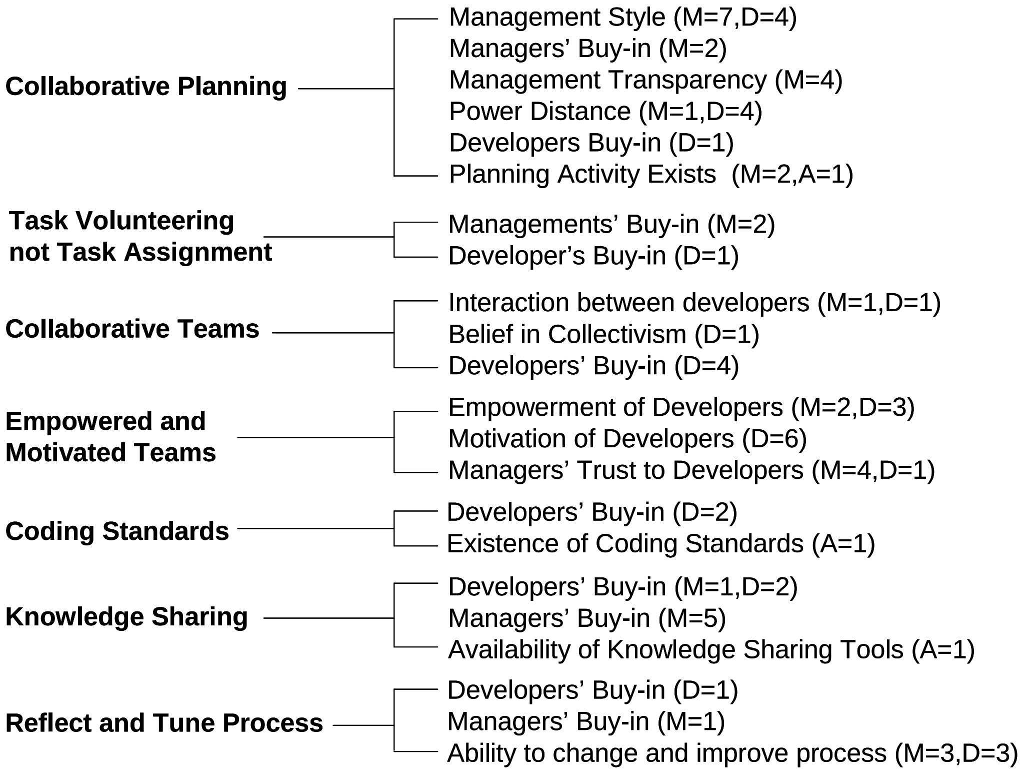

Collaborative Planning
Management Style (M=7,D=4)
Managers' Buy-in (M=2)
Management Transparency (M=4)
Power Distance (M=1,D=4)
Developers Buy-in (D=1)
Planning Activity Exists (M=2,A=1)
Task Volunteering not Task Assignment
Managements' Buy-in (M=2)
Developer's Buy-in (D=1)
Collaborative Teams
Interaction between developers (M=1,D=1)
Belief in Collectivism (D=1)
Developers' Buy-in (D=4)
Empowered and Motivated Teams
Empowerment of Developers (M=2,D=3)
Motivation of Developers (D=6)
Managers' Trust to Developers (M=4,D=1)
Coding Standards
Developers' Buy-in (D=2)
Existence of Coding Standards (A=1)
Knowledge Sharing
Developers' Buy-in (M=1,D=2)
Managers' Buy-in (M=5)
Availability of Knowledge Sharing Tools (A=1)
Reflect and Tune Process
Developers' Buy-in (D=1)
Managers' Buy-in (M=1)
Ability to change and improve process (M=3,D=3)

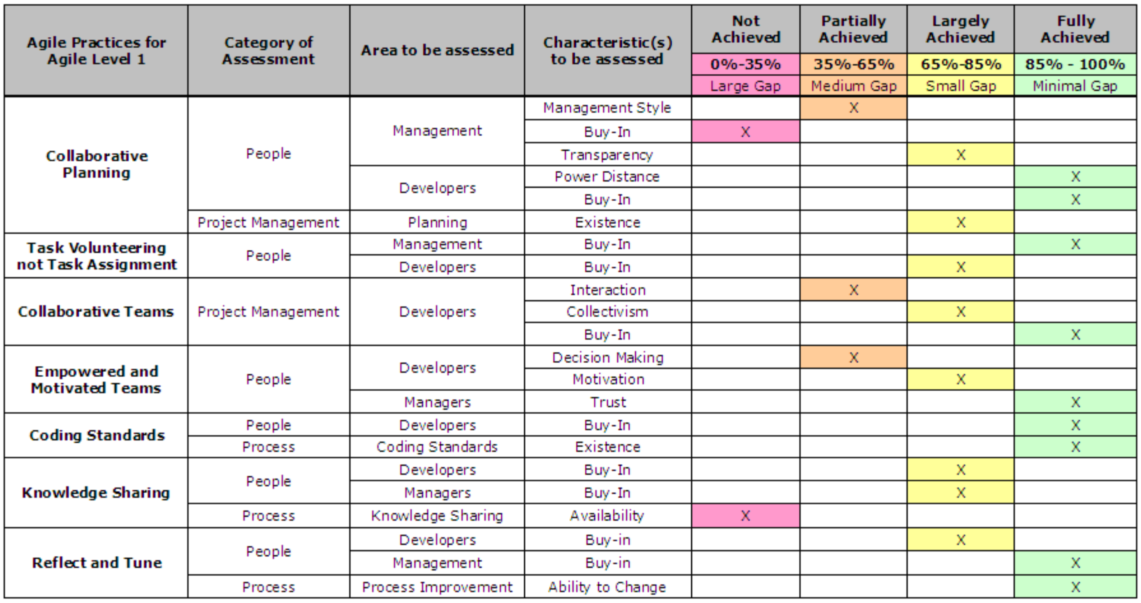
Figure 12. Organizational Characteristics Assessed for Agile Practices in Level 1

The fundamental difference between limiting and non-limiting agile practices is the issue of whether the organization has control over changing the characteristics needed for the adoption of that agile practice. With limiting agile practices, the organization does not have the ability to change these characteristics and, therefore, they limit the highest level of agility the project can reach. However, with non-limiting practices, the organization *can* change them. Therefore, the focus becomes determining if the current state of these characteristics can support adopting the agile practice or not. Since they are non-limiting, if the assessment results show these characteristics are not in a state to support the adoption of an agile practice, the organization can strengthen and change these organizational characteristics until they are ready to accommodate the agile practice.

Assessing the Organization's Readiness

Given an identified project, the first step is to determine the set of agile practices or Candidate Practices the organization aspires to adopt. To save time and cost during this assessment stage, instead of assessing how ready the organization is to adopt all the agile practices in the SAMI, the set of candidate practices consist of only the practices within the target agile level of the project and the levels below that. For example, if the target agile level for a project is Agile Level 3, then the set of candidate practices comprises all the practices in Agile Levels 1, 2 and 3. These are candidate practices, because they are the practices that the project *wants* to adopt, but is waiting for the results of the organizational readiness assessment to determine which ones the project *can* actually adopt. Once the assessor has identified the candidate practices, he or she uses the indicators associated with each agile practice (see Chapter 4) to determine the extent to which the organization is ready to adopt that practice. Figure 12 shows the organizational characteristics that the assessor needs in order to assess the agile practices in Level 1. The assessor uses multiple indicators to assess each of these organizational characteristics. The results for each of these characteristics are aggregated together using an approach inspired by the Evaluation Environment [10] and are plotted in a table similar to the one in Figure 13.

Figure 13 shows the table resulting from the organizational assessment stage that depicts the achieved level of each organizational characteristic. The highest level of aggregation for the indicators is that of the organizational characteristic. If the results were aggregated up to the level of the agile practice, the results would be useless. Knowing that an organization is "partly" ready for an agile practice is not beneficial. However, keeping the level of aggregation at the characteristic level is beneficial to executives and decision makers, because it draws attention to the characteristics of the organization that might cause the adoption of a practice to fail.



Agile Practices for Agile Level 1	Category of Assessment	Area to be assessed	Characteristic(s) to be assessed	Not Achieved	Partially Achieved	Largely Achieved	Fully Achieved
				0%-35%	35%-65%	65%-85%	85% - 100%
				Large Gap	Medium Gap	Small Gap	Minimal Gap
Collaborative Planning	People	Management	Management Style		X		
			Buy-In	X			
			Transparency			X	
		Developers	Power Distance				X
			Buy-In				X
	Project Management	Planning	Existence			X	
Task Volunteering not Task Assignment	People	Management	Buy-In				X
		Developers	Buy-In			X	
Collaborative Teams	Project Management	Developers	Interaction		X		
			Collectivism			X	
			Buy-In				X
Empowered and Motivated Teams	People	Developers	Decision Making		X		
			Motivation			X	
		Managers	Trust				X
Coding Standards	People	Developers	Buy-In				X
	Process	Coding Standards	Existence				X
Knowledge Sharing	People	Developers	Buy-In			X	
		Managers	Buy-In			X	
	Process	Knowledge Sharing	Availability	X			
Reflect and Tune	People	Developers	Buy-in			X	
		Management	Buy-in				X
	Process	Process Improvement	Ability to Change				X

Figure 13. Organizational Readiness Assessment Results

As in the project level assessment, determining the highest agile level an organization is capable of achieving is dependent on the organization's readiness to adopt the practices in that agile level. If the organizational characteristics needed for a practice are found to be *not achieved* or only *partially achieved*, this finding indicates that the organization is not ready to adopt that practice. As a result, the highest level of agility the organization can reach becomes the level at which a necessary organizational characteristic is missing. For example, as Figure 13 shows, since *collaborative planning* is in Agile Level 1, and since two of the characteristics that it needs are deficient (management style is *partially achieved* and management buy-in is *not achieved*), the highest level of agility for that organization is Level 1.

Stage 3 ends once the results of the organizational readiness assessment are plotted. Stage 4 provides guidelines for analyzing the results and developing an action plan based on these results.

3.6. Stage 4: Reconciliation

Following the organizational readiness assessment, the agile level achievable by the organization is known. Prior to that, Stage 2 had identified the agile level that the project aspires to adopt. Therefore, the final step, reconciliation, is necessary to determine the agile practices the project finally adopts. In essence, during this stage the assessor analyzes the results of the organizational assessment and makes a set of recommendations to the organization on how to proceed, especially if the organization's readiness level is less than the project target level.

Figure 14 illustrates the activities of this stage. A simple gap analysis examines the difference between the organizational readiness level and the project target level. If there is no gap, meaning that the organization is ready to achieve the project's target agile level, then the organization has successfully identified the set of agile practices to adopt. However if there is a gap between the organization's readiness level and the project's target level, then reconciliation is needed. As explained in detail later, the reconciliation phase will either recommend the strengthening of weak organizational elements or to adopt those practices the organization is ready for, even though the project won't be operating at its full agile potential.

During Stage 4 the differences between the project's target agile level and the organization's readiness level are resolved to determine the final set of agile practices to be adopted/employed. Three different scenarios are possible during this stage:

- *Organization Readiness Level is higher than the Project Target Level:* This case is depicted conceptually in Figure 15. No reconciliation is needed and all the practices within the project's agile level and below become the chosen agile practices for adoption. This scenario rarely occurs, because the project environment is usually contained within the organization.

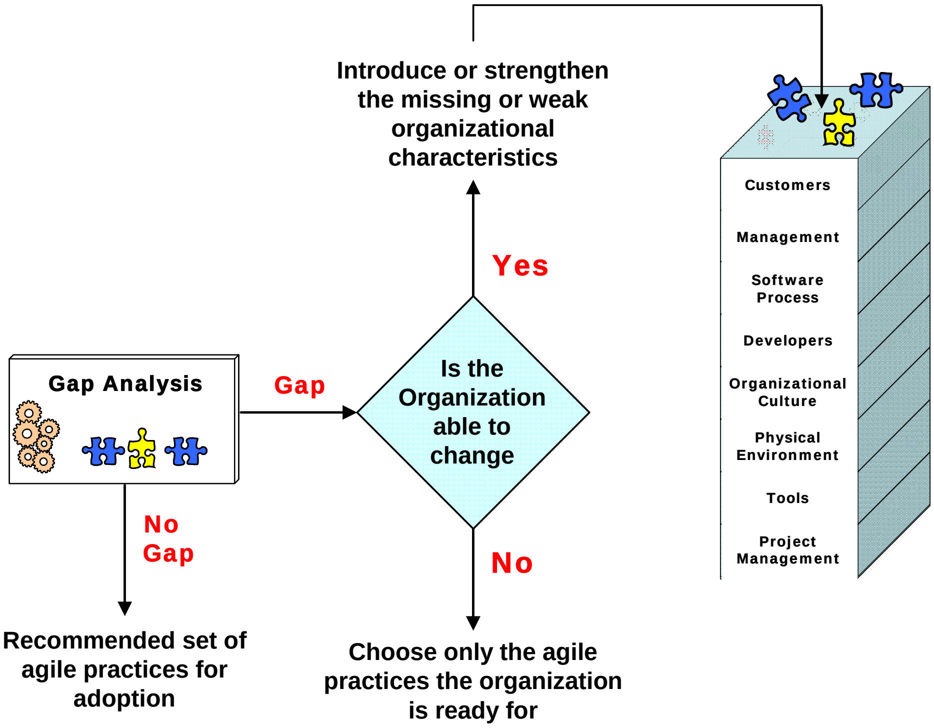

Introduce or strengthen the missing or weak organizational characteristics
Yes
Gap Analysis
Gap
Is the Organization able to change
No Gap
No
Recommended set of agile practices for adoption
Choose only the agile practices the organization is ready for
Customers
Management
Software Process
Developers
Organizational Culture
Physical Environment
Tools
Project Management

Figure 14. Stage 4: Reconciliation

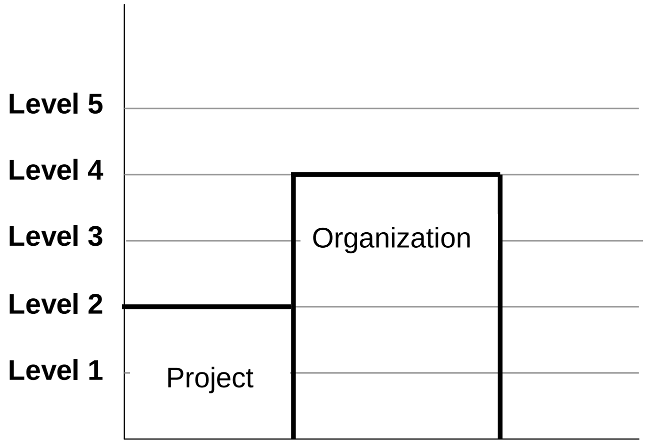


Figure 15. Reconciliation (Organization > Project)

- *Organization Readiness Level = Project Target Level:* This case is depicted conceptually in Figure 16. No reconciliation is needed and all the practices within the project's agile level and below become the agile practices chosen for adoption. This is the ideal case, because the project is achieving 100% of its agile potential.

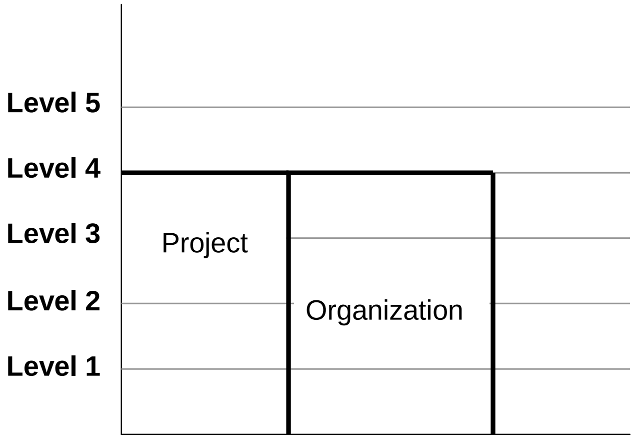


Figure 16. Reconciliation (Organization=Project)

- *Organization Readiness Level < Project Target Level:* As depicted in Figure 17, reconciliation is necessary. Stage 4 provides two options for reconciling this situation. Each option is presented below.

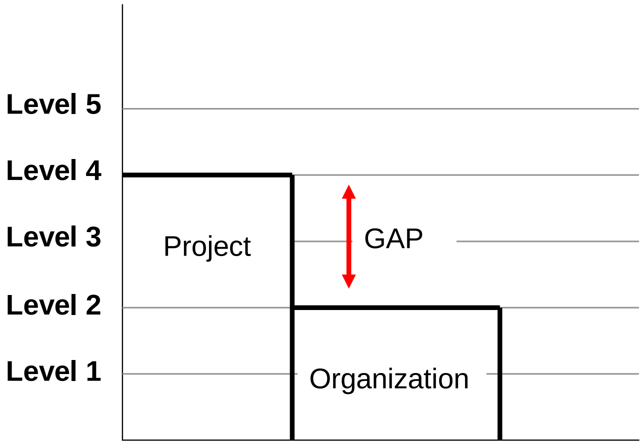
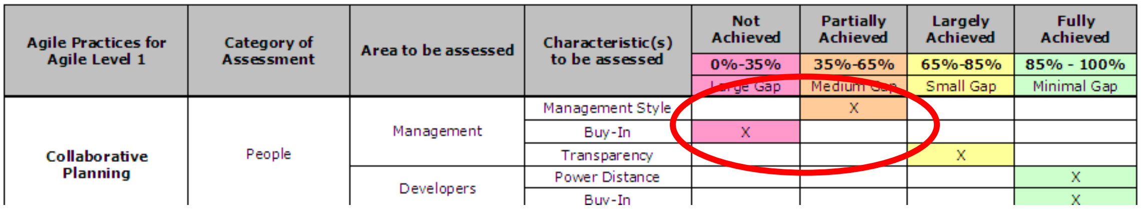


Figure 17. Reconciliation Stage (Organization < Project)

Option 1: Change the Organization

The first option relies on how ready and willing the organization is for changes and improvements. The results of the organizational assessment have identified the exact characteristics hindering the organization from reaching higher levels of agility (i.e. the project's target level). If changing any of these characteristics is within the control of the organization, then the organization can undertake the necessary steps to improve these characteristics.

While the Agile Adoption Framework does not include a list of best practices for improving or strengthening the organizational characteristics found to be weak, the framework provides enough guidance for the organization to find resources to improve these weaknesses. For example, Figure 18 shows that management style and Buy-in are the two factors keeping the organization from being able to adopt Collaborative Planning. Although how to fix these challenges is not within the scope of the framework, a simple Internet search provides many resources to improve these characteristics [65] [47] [80] [86]. Changing some of these organizational characteristics might be as simple as buying a software tool or as difficult as orchestrating a complete cultural change. Reading books and applying some of the principles mentioned in them is one way to strengthen these organizational characteristics. In other cases, such as the one Figure 18 indicates, the managers might have to go through some kind of training to change their mindset and promote a cultural change within the organization. Every organization must find its own approach to overcoming the weakness identified from the organizational assessment.



Agile Practices for Agile Level 1	Category of Assessment	Area to be assessed	Characteristic(s) to be assessed	Not Achieved	Partially Achieved	Largely Achieved	Fully Achieved
				0%-35%	35%-65%	65%-85%	85% - 100%
				Large Gap	Medium Gap	Small Gap	Minimal Gap
Collaborative Planning	People	Management	Management Style		X		
			Buy-In	X			
			Transparency			X	
		Developers	Power Distance				X
			Buy-In				X

Figure 18. Part of the Organizational Readiness Assessment results

While Stage 4 does not offer a specific action plan for fixing the problems identified, it provides the organization with a certain level of guidance on what needs to be done and in what order. The framework suggests that the order of the changes should follow the roadmap provided by the SAMI. In other words, the organization should try to fix all the challenges pertaining to the agile practices in level 1 before it moves to the challenges related to the practices in level 2 and so on. It is because of this stage that Section 2.3 mentioned that the 4-Stage process also addresses some of the objectives of the Establishing phase in IDEAL. This is because this stage provides the organization with suggestions concerning the priority by which change recommendations should be implemented.

When an organization has made all the recommended changes, it can support agile practices at the project's target level. However, if the organization is not ready for change, then the second option is put into action.

Option 2: Lower Expectations

The second option is suitable for organizations unwilling to invest time, effort or money to make changes, or are unable to change some of the weaknesses identified from the organization assessment. By lowering their expectation, these organizations can opt to adopt only the agile practices that are within their current capacity. The priority of such an organization should be to focus on adopting all the practices it can that are within the same agile level. The reason for this is that the practices in the same agile level are grouped together to create a certain synergy when adopted together. Therefore, the organization needs to take advantage of this synergy and try to completely adopt the practices in one level before going to the next. The obvious downside to this option for reconciliation is that the project is restricted to operating at a lower level of agility than its potential.

This reconciliation stage helps the organization identify the agile practices it can realistically adopt. At the same time, if the organization is able and willing to improve, then this stage guides it to where the improvements need to occur to enable the project operates at its full agile potential. Moreover, by utilizing this approach, the organization prepares itself sufficiently before starting the process of introducing agile practices into the development process, thereby decreasing the impact of the adoption process.

The final product of the 4-Stage Process is a set of recommended agile practices that are suitable for the organization to adopt. How the actual adoption is achieved is outside the scope of this research. For each agile practice the organization is ready to adopt, either a specialized consultant is hired to introduce a particular agile practice to the project or the project team can just read a specialized book about the agile practice they plan to adopt. Most of the agile practices have one or more dedicated books that explain them in detail.

The main contribution of the 4-Stage Process along with the SAMI is to provide organizations with guidance on how to start the agile adoption process and which agile practices they should adopt. Moreover, the framework provides detailed recommendations on what the organization needs to improve on to successfully adopt its desired agile practices.

It is evident from this chapter that the 4-Stage Process component of the Agile Adoption Framework relies heavily on the Sidky Agile Measurement Index (SAMI). The next chapter presents, in detail, the structure of the SAMI and how it is structured.

4. The Sidky Agile Measurement Index

Chapter 3 highlights the 4 Stage Process, the main component of the Agile Adoption Framework. The first stage of this Process helps determine whether organizations are ready to undergo agile adoption efforts. The second and third stages provide a means for projects and organizations to assess their agile potential using the Sidky Agile Measurement Index (SAMI). The last stage, Stage 4, suggests a final set of Agile Practices for organizations to adopt by reconciling any differences between the Agile Levels identified in Stage 2 and Stage 3. The SAMI is instrumental in identifying the highest level of agility a project can reach (the goal of Stage 2), identifying the level of agility the organization is ready to adopt (the goal of Stage 3), and reconciling any existing differences (the goal of Stage 4). This chapter is dedicated to presenting the details of the SAMI.

This chapter begins with Section 4.1, which discusses background information about measurement indices both in general and as related to agility. Each of the subsequent 4 sections presents a different component of the SAMI. Section 4.2 discusses the notion of Agile Levels in detail, including the process of their creation and varying levels of significance. Section 4.3 presents the role of Agile Principles and their importance in the measurement index. In Section 4.4, a comprehensive example describes how each Agile Level is populated with Agile Practices with the help of Agile Principles. Section 4.4 also includes the taxonomy and description of each of the Agile Practices. The fourth component of the SAMI, the Indicators, is presented in Section 4.5. The final section of this chapter, Section 4.6, discusses the tailorability of the SAMI.

4.1 Background Information about Measurement Indices

Before developing a measurement index capable of measuring *agile potential* relative to the core values (objectives) of agility, it was necessary to find out whether an adequate one already existed. We started first by looking at popular measurement indices in Software Engineering to decide whether or not it was adequate.

The first candidate was the CMMI, a well-defined and widely accepted measurement index for software development processes. While analyzing the CMMI -and the Key Process Areas (KPAs) it assesses- to determine its capability, it became apparent that CMMI was not suitable for measuring the agility of a process. CMMI is designed to measure process maturity and capability, not agility. While it is impossible to use CMMI to assess agility, it is possible to use a number of its structural aspects to design an agile measurement index. These aspects are highlighted later in further detail.

While there is no agile measurement index able to determine the agile potential of a project, there have been other attempts to create so-called measurement indexes for agility. However, most of what is published on the subject of agile measurement takes the approach of determining a suitable *process methodology* for a specific type of project rather than finding the *right degree of process agility* for a project. The approaches involved with determining suitable process methodologies look at the whole "planning spectrum" (see Figure 19) and attempt to identify, in a non-pragmatic manner, which agile process methodology is most suitable for the given project.

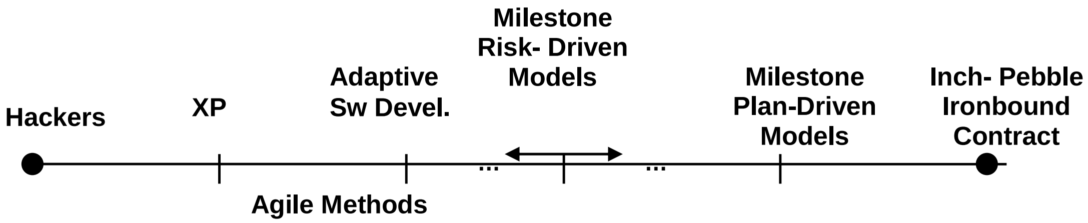

Hackers
XP
Adaptive Sw Devel.
Milestone Risk- Driven Models
Milestone Plan-Driven Models
Inch- Pebble Ironbound Contract
...
...
Agile Methods

Figure 19. Planning Spectrum [19]

In his book *Balancing Agility and Discipline* [22] Barry Boehm argues for the use of a risk-driven approach for determining the right level of planning necessary for a project. He makes use of different types of risks (e.g. environment risks, agility risk, and plan-driven risks), coupled with different project characteristics (criticality, personnel, rate of change for the requirements, team size and culture). With these he tells how to find the "sweet spot," or the most suitable level of process definition for the organization. Boehm's research, however, fails to translate this sweet spot into actual Agile Practices. Consequently, once the right balance of agility and discipline is defined, it must be established: what the right balance means for an organization, what this balance should do, and which Agile Practices are equal to the operational level of agility. While Boehm's research on how to balance agility and discipline is informative, well documented, and validated, it lacks guidelines for an organization adopting agility on which steps and practices are necessary to reach the identified level of agility.

Other work, bearing a name similar to the SAMI, is the Agility Measurement Index (AMI) [33]. The AMI also serves as a heuristic approach for deciding which methodology is best for a given project. The AMI defines five dimensions of a project that, when calculated, define the Agility Measurement Index of a project. These five dimensions are Duration, Risk, Novelty, Effort and Interaction.

The problem with Boehm's research, the AMI and similar approaches is that they fail to identify the agile potential of a project. Instead, they recognize the existence of a planning spectrum (different levels of planning) and try to find the ideal degree of "process planning" for a project. Moreover, in some cases (e.g. mission and life critical systems) these approaches might suggest a non-agile process as the ideal process for the development of these systems. This is a point that is difficult to accept because every project, even mission and life critical systems, can adopt some level of agility [78].

Another observation about the planning spectrum is the way the spectrum is structured. If this spectrum is thought of as a “measurement index” for how agile a process is, it becomes evident that the units of this measurement index are actually software development methodologies. For example, Figure 19 shows that XP is a “unit” on this measurement index and Adaptive Software Development (ASD) is another unit on the other side of the “agile spectrum.” Having specific agile methodologies as “units” of the measurement index, and using this measurement index results in processes being measured according to their *adherence level* to that specific agile methodology. An agile measurement index must use agile values (i.e. objectives of agility) as units, not specific agile methods. The CMMI is based on values and Key Process Areas, not on specific software development methodologies. The Waterfall development model is not CMMI Level 1 and the Spiral model is not CMMI Level 2. The levels of CMMI (i.e. the units of measurement for CMMI) are independent of any particular development model or methodology. However, for some reason when it comes to agility some of the measurement approaches use specific agile methods as the units.

Although the best existing approaches, highlighted above, offer useful contributions to developing an agile measurement index, none of them are suitable for identifying the agile potential of a project and assessing the readiness of an organization for agility. Therefore, after the search for a suitable agile measurement index failed, the need to create a new one became evident. The next section discusses in detail the first component of the Sidky Agile Measurement Index (SAMI) – Agile Levels.

4.2 Agile Levels

Since the objective is to create a measurement index to measure organizations adopting agility, a fundamental question needs to be answered before determining how to measure agile potential; how does an organization move towards and adopt agility. After much thought, the answer is obvious; organizations become more agile when they adopt more Agile Practices. As a result, in very general terms and on a very macro level, the more Agile Practices an organization adopts, the more agile it is. An organization that has adopted 10 Agile Practices is most probably more agile than an organization that uses fewer or none at all. This fundamental concept helps answer the critical question of how to measure agile potential – by the number of Agile Practices software development process can adopt. As a result, the Sidky Agile Measurement Index (SAMI) is designed to measure the agile potential of an organization using the notion of Agile Practices.

Once it is determined how to measure the agile potential, the next crucial question relating to the SAMI is what units this measurement index will use to measure agile potential. A temptingly simple solution is to count the number of Agile Practices an organization can adopt and make the sum its "agile potential score." However, this approach is too simplistic and inaccurate. For example, it is inaccurate to say an organization able to adopt five Agile Practices has more agile potential than an organization that can adopt four practices, because many other factors must be considered, including the type of practices employed and the impact of each practice on the organization's agility. The latter is especially important, because not all Agile Practices have the same level of impact on an organization's agility

When a simple count of Agile Practices proves inadequate, the search for a solution continues with looking at other process improvement measurement indexes for inspiration. The CMMI and other process improvement standards and measurement indexes, such as the ISO 15504 (SPICE), used *levels* as units for their measurement indexes. CMMI has six different maturity levels ranging from 0 to 5 and SPICE has six capability levels.

The notion of levels in CMMI and SPICE inspired the decision to make the units for the SAMI *Agile Levels*. The next challenge is to define the nature of these levels within the agile measurement index. An explanation of this process follows in the next sections.

With the agile measurement index measuring the agility of an organization by the use of Agile Practices with Agile Levels as its units, it becomes necessary to find the number of levels needed and how to define each level with respect to Agile Practices. Again the CMMI provides help. In the CMMI, each maturity level stabilizes an important part of the organization's processes, and each maturity level comprises a predefined set of process areas. Every process area consists of a cluster of related practices that, when performed collectively, satisfy a set of goals considered important for making significant improvement in that area. Similarly, each agile level introduces an important agile quality (e.g. collaboration) into the organization to help it become more agile. Each agile level consists of a set of Agile Practices that are related and, when adopted collectively, make significant improvements in the software development process, thereby leading to the realization of the agile quality of the respective agile level.

4.2.1 Identifying the Levels of Agility

The next challenge is to define the levels of agility. The Agile Levels must be independent of any particular agile method, so it is not suitable to have agile level 1 be Adaptive Software Development and agile level 2 be Extreme Programming. The levels must be based on the core values and qualities of agility. For this reason, the starting point for defining the Agile Levels of the SAMI is the Agile Manifesto, the original document that highlights the core values and principles of agile software development, which states:

> We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:
>
> - Individuals and interactions over processes and tools
> - Working software and over comprehensive documentation

- Customer collaboration over contract negotiation
- Responding to change over following a plan

That is, while there is value in the items on the right (e.g. Individuals and interactions), we value the items on the left (e.g. processes and tools) more.

The original text of the Agile Manifesto, as well as the 12 principles developed from the manifesto and the review of the body of literature related to agility helped identify five values or qualities of agility. They are:

- **Responding to change through multiple levels of feedback:** The whole paradigm shift towards agile pivoted around this goal of responding to change. Agile development ensures that multiple levels of feedback are present to enable the notion of adapting to change [45].
- **Ensuring continuous delivery of software:** The value of early and continuous delivery of software is fundamental to agile software development Every agile method is based on the notion of evolutionary development (i.e. small and continuous increments) [60].
- **Establishing communication and collaboration:** This value is concerned with fostering communication and collaboration between all the different stakeholders. Collaboration is the foundation of agile software development [80] [24] [27].
- **Producing quality software:** Agile processes seek to employ engineering practices that foster the development of quality software. In order for an agile process to be nimble and adapt quickly to change, effective engineering practices must support the technical process [51] [27].
- **Providing an all-encompassing agile environment:** Agility is not only a set of practices, but also essentially a culture. Therefore, it is important to have an environment that is reflective and supportive of the agile nature of the software development process.

After identifying these key agile values and qualities, the task of converting them into Agile Levels is next. Once the need for the five levels -each representing one of the agile values- became evident, the next step is to name and sequence the levels. Table 9 shows the names given to each level and the agile value or quality each represents

Agile Level Name	Agile Value or Quality
Adaptive	Responding to change through multiple levels of feedback:
Evolutionary	Ensuring continuous delivery of software
Collaborative	Establishing communication and collaboration
Effective	Producing quality software
Encompassing	Providing an all-encompassing agile environment

Table 9. Agile values represented by Agile Levels

4.2.2 Determining the Order of the Levels of Agility

Once named, the levels have to be carefully sequenced in a manner that provides a roadmap for those aspiring to move toward agility. This roadmap answers the questions of which agile level to introduce first and why.

The search for the most appropriate sequence for the levels involves reviewing the Agile Manifesto, various agile books and articles, organizational change books and even books about social change and its causes. One book in particular stands out, *The Tipping Point* by Malcolm Gladwell [42]. Even though Gladwell's book discusses mainly the phenomena of social outbreaks (whether positive or negative) and how they are caused, it also provides a general framework for the sequencing of the Agile Levels in the SAMI. Gladwell explains how fashion starts and spreads and how crime waves start and die out using three main laws or factors. The first law, the "Law of the Few", highlights the role of people in the spread of social changes. Detailing the different kinds of people needed for these social outbreaks, Gladwell demonstrates that the first factor is all about people. The second law, “the stickiness factor,” is concerned with the actual content of the message or of the concept being spread. The last factor, which Gladwell calls “the context,” is all about the environment. Gladwell shows that to complete a social change or outbreak it is necessary to setup an environment or context that is symbolic of this message or concept.

This analysis of *The Tipping Point* led to an epiphany. The five Agile Levels shown in Table 9 follow the same generic framework. The first factor in this book focuses on people and relationships, as does the agile level named “collaborative”. The last factor, “the context” matches perfectly the agile level named “encompassing,” which focuses on establishing an all-encompassing environment. The middle factor in this book, the content of the message or concept, matches the remaining three Agile Levels that were related to the actual nature of the agile development process, or in other words, the content.

These realizations set the basis for the sequencing of the Agile Levels. The first level (Agile Level 1) focuses on establishing communication and collaboration. The last agile level (Agile Level 5) focuses on providing an all-encompassing agile environment. For the rest of the levels it must be determined which agile value should be introduced first into an organization. This elicits two more concerns to be determined: which of these values would have the biggest impact on moving an organization towards agility, and if the agile values dependant upon each other. The obvious answer to both is that Agile Level 2 ensures early and continuous delivery of software. The basis of this decision is that most of the Agile Practices are dependant on the fact that the development is conducted in an evolutionary manner rather than the big bang approach. As a result, the effective engineering practices cannot be introduced first, because they depend on an evolutionary development process. Moreover, the agile value of responding to change pivots around the use of evolutionary development. Once this is decided, it was obvious that Agile Level 3, the effective level, should focus on producing quality software. The reason for this is that sound technical practices that enable the process to produce working quality software are a prerequisite to having the ability to respond to change. Before the product can quickly adapt to constant changes, it must make sure that no changes will jeopardize the quality of the product. Hence, the effective agile level precedes the adaptive level, or Level 4. Table 2 displays the sequence of the Agile Levels in the measurement index. The levels are shown in reverse order to reflect the notion that agility increases with each agile level of attainted.

Agile Level	Level Name	Level's Objective (Agile Value Re-worded)
Level 5	Encompassing	Establishing a vibrant and all-encompassing environment to sustain agility
Level 4	Adaptive	Responding to change through multiple levels of feedback
Level 3	Effective	Developing high quality, working software in an efficient an effective manner
Level 2	Evolutionary	Delivering software early and continuously
Level 1	Collaborative	Enhancing communication and collaboration

Table 10. The 5 Agile Levels in order

Agile Level 1: Collaborative or Evolutionary?

After organizing the Agile Levels there is some question about the first two levels in particular. Since the Agile Levels now represent a roadmap for an organization moving towards agility, the levels this suggest the steps this organization should take to move toward agility. The disputed questions arise over whether the first step should be about communication and collaboration or about ensuring there is an evolutionary development process. While this is a legitimate concern, however, ultimately it would still yield a higher impact towards agility for the organization if the move to agility began with communication and collaboration.

The primary reason for the decision to have communication and collaboration as the first level is the first sentence from the Agile Manifesto itself:

Individuals and interactions over processes and tools

If the first step towards agility focused on changing the software development process, and not enhancing communication and collaboration between the individuals, it would be contradicting the agile manifesto itself.

The CMMI and the traditional process improvement paradigm provide another reason for making communication and collaboration the focus of Level 1. In 1993 Watts Humphrey developed the Personal Software Process (PSP) and its companion the Team Software Process (TSP) [48]. These software development processes were a set of guidelines and best practices to incorporate discipline in the software development process. What proves interesting, especially with TSP, is its focus on enhancing communication and collaboration. Success stories have highlighted the contribution that TSP has had on improving the software development efforts of various organizations [20]. Therefore, it is obvious, even within the CMMI paradigm, that adding communication and collaboration practices would enhance the process tremendously.

These two reasons, the opening sentence of the Agile Manifesto and the observations from the TSP, lead to the conclusion that enhancing communication and collaboration should be the first step for an organization's move toward agility. In 2006, Alistair Cockburn, one of the original authors of the Agile Manifesto, published the second edition of his book about agile software development, *Agile Software Development: The Cooperative Game* [25]. The title's wording confirmed Agile Level 1 as collaborative. In Chapter 3 (First Who, Then What) of his book *Good to Great: Why Some Companies Make the Leap... and Others Don't,* Jim Collins also focuses on building people first [32]. Just as this literature offers much evidence that collaboration should be the first step and priority of software development [27, 49, 45], the title of Kert Peterson's article in 2005 "*Collaboration: The key to Enterprise Agile Adoption"* [70], provides yet another sign that the focus of Level 1 should be enhancing communication and collaboration. In fact, the literature on agile adoption underscores the notion that collaboration truly is the first step to success in the agile adoption process.

When is an organization considered "Agile"?

The above conclusion brings up the question of how to decide when an organization is agile. For example, is an organization considered agile even if it is only at Level 1? What if an organization is still using the waterfall model, and adopts all the Agile Practices in Agile Level 1, is this organization considered "agile"? These questions are about titles and names, not substance. It adds no fame or glory to call an organization agile when it is not. The main point is that no one can deny enhancing communication and collaboration in an organization is a step towards *agility* (See Figure 20). The concern here is with moving organizations towards agility, not with giving them certificates and recognition when they become agile.

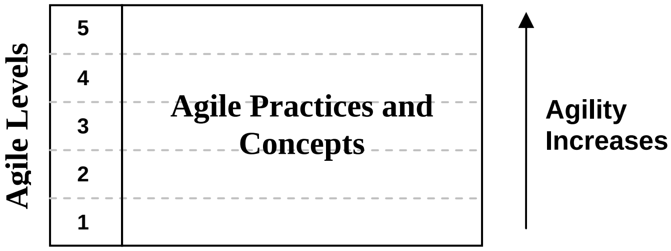


Figure 20. Agile Levels

Since the measurement index uses Agile Practices to measure agility, each of the Agile Levels must consist of a set of different Agile Practices. The next step in the development of the agile measurement index is to populate each of the levels with relevant Agile Practices. The role of the second component of the SAMI, the Agile Principles, is to provide organizations with guidance on how to properly populate each agile level with the right set of Agile Practices. The adherence to Agile Principles ensures that each the software development process is agile. The next section discusses what Agile Principles are, and how they are used to develop the SAMI.

4.3 Agile Principles

Agile principles are the essential characteristics that need to be embodied in a process before it is considered agile. When writing the Agile Manifesto, the authors identified 12 principles for agile software development [2].

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity--the art of maximizing the amount of work not done--is essential.

11. The best architectures, requirements, and designs emerge from self-organizing teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

In terms of the development of the SAMI, knowing the principles of agility is important because they play a key role in populating the Agile Levels with Agile Practices.

4.3.1 The Relation between Agile Levels and Agile Principles

An analogy that helps illustrate the relation between Agile Principles and Agile Levels takes its inspiration from Wake's "Slicing the cake" analogy used for dividing user stories [82]. Think of a development process that is fully agile as a multi-layer cake, where each layer of the cake is an agile principle, thereby making this 12-layer cake, and that Agile Levels are slices of the cake. To make sure that each slice of the cake (Agile Level) exemplifies the essence of the whole cake (agility), all the layers of the cake (Agile Principles) should be present in each slice.

The same concept holds true for the SAMI. To ensure that the Agile Levels embody the essential characteristics of agility, each level is created by adhering to as many, if not all, of the Agile Principles. The makeup of each agile level, in terms of Agile Practices, is guided by the level's scope of adherence to these Agile Principles. Each agile level adheres to the Agile Principles for different purposes, depending on the agile value the level is introducing into the process.

The reason each agile level needs to exemplify all of the Agile Principles is to ensure that when an organization adopts the Agile Practices within any given agile level it is not only adopting one aspect of agility (one layer of the cake). If it were to adopt only one aspect of agility the software development process after the adoption of that agile level will not exhibit an agile behavior. Moreover, if each agile level were to focus on only one or two Agile Principles, then the roadmap to agility is not agile in and of itself, because it would be following a waterfall approach where the whole product (agility or an agile process) would only come into existence at the end of the whole process. Agile promotes itself using the fact that after each iteration there is a potentially shippable product, which means that at each iteration the product consists of all of its levels and does not wait to come together only at the end of the development process.

4.3.2 Identifying the 5 Agile Principles used in the SAMI

The 12 Agile Principles are used as guidelines for populating the Agile Levels with Agile Practices. However, if all of the 12 Agile Principles are used to define each agile level, unnecessary complications will be introduced. Careful grouping and summarization make it possible to identify the five Agile Principles that capture the essence of all 12 principles. These five Agile Principles guide the development of the 5 Levels of Agility:

- *Embrace change to deliver customer value* [17]**.** The success of a software development effort is based on the extent to which it helps deliver customer value. In many cases, the development team, as well as the customer, are in a continuous learning process as to the requirements necessary to realize additional customer value. Hence, an attitude of welcoming and embracing change should be maintained throughout the software development effort.
- *Plan and deliver software frequently* [18] [29] [72]**.** Early and frequent delivery of working software is crucial, because it provides the customer with a functional piece of the product to review and provide feedback on. This feedback is essential for the process of planning for upcoming iterations, as it shapes the scope and direction of the software development effort.
- *Human-centric* [24]**.** The reliance on people and the interactions among them is a cornerstone in the definition of agile software processes.

- *Technical excellence* [45] [57]**.** Agile developers are committed to producing only the highest quality code possible, because high quality code is essential in high-speed development environments, such as the ones characterized as agile.
- *Customer collaboration* [18]**.** Inspired by the original statement of the agile manifesto, there must be significant and frequent interaction between the customers, developers, and all the stakeholders of the project to ensure that the product being developed satisfies the business needs of the customer.

In effect, Agile Principles are used to ensure that each Agile Level embodies essential characteristics of agility. Figure 21 illustrates the relationship between Agile Levels and Agile Principles.

Each agile level should contain Agile Practices associated with most, if not all, of the Agile Principles. The principle reflects the approach that the agile practice uses to promote the agile quality pertinent to a level. For example, since adopting the practices within agile Level 1 renders a collaborative process. By adhering to the technical excellence principle during the creation of that level, Agile Practices and concepts that are related to technical excellence become part of the set of practices that makeup the level. Therefore, even when the agile level is focused on collaboration, it will still contain practices that exhibit technical excellence, and at the same time, promote collaboration. As for Agile Level 3, it will contain practices that exhibit technical excellence but promote its agile value – effectiveness.

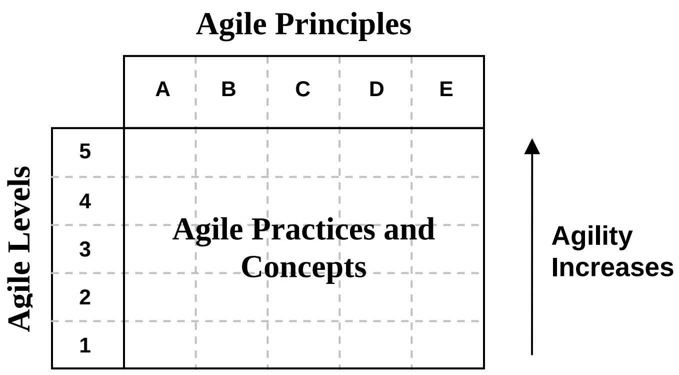


Figure 21. Layout of Agile Levels and Principles within SAMI

At this point, the agile measurement index has two dimensions, the Agile Levels and the Agile Principles. However, the measurement index is empty, as Figure 22 shows, and cannot be used yet. To complete the rendition of the agile measurement index, the Agile Levels, by adhering to these five Agile Principles, are populated with Agile Practices and concepts. The next section provides an explanation of how the agile measurement index is populated with the Agile Practices and concepts.

	Agile Principles				
	Delivering Customer Value by Embracing Change	*Planning to Deliver Software Frequently*	*Human-centric*	*Technical Excellence*	*Customer Collaboration*
Level 5: **Encompassing**					
Level 4: **Adaptive**					
Level 3: **Effective**					
Level 2: **Evolutionary**					
Level 1: **Collaborative**					

Figure 22. Empty Matrix of the 5 Agile Levels and 5 Agile Principles

4.4 Agile Practices

The previous two sections explain Agile Levels and Agile Principles and the relationship between them. Agile Levels are the basic measurement unit of the SAMI and are used to assess the agile potential of a project or organization. Each Agile Level is composed of a synergistic set of Agile Practices that, when adopted, cause the software development process to realize a new value or quality of agility. Agile Principles guide the process of populating each Agile Level with the appropriate set of Agile Practices. Agile Practices are the concrete activities and practical techniques used to develop and manage software projects in a manner consistent with the Agile Principles. At the same time, each Agile Practice is categorized under the Agile Principle that it manifests. Examples of popular Agile Practices include *paired programming*, *user stories,* and *collaborative planning*.

Industry uses many of the currently known Agile Practices even as new Agile Practices are being developed. Some Agile Practices are borrowed from other disciplines and some are created to meet the special development needs of the agile community. In either case, Agile Practices either replace "non-agile" practices or redefine or complement them. For example, user stores are promoted as a replacement for the traditional requirements specification document and Test Driven Development (TDD) complements current or traditional development practices. Paired programming is either a new practice or can be considered a redefinition of how programmers should work. Although there are advocates and critics of many of these practices, it is not within the scope of this research to comment on these practices.

Before populating each Agile Levels in the SAMI with its suitable set of Agile Practices, a list of all the Agile Practices needs to be known. The approach taken to identify all the Agile Practices is to collect all Agile Practices used by current agile development methodologies [51] [57] [3]. Because of the large number of redundancies, the collection process includes grouping similar practices together. Also, some of the names of the practices are changed to reflect a more generic approach to the practice. After these adjustments, the result is a set of 40 different Agile Practices. This compilation provides a starting point, but does not by any means limit Agile Practices to this list only.

The Agile Practices are grouped according to the Agile Principles identified in Section 4.3. Each of the practices listed used to exemplify and manifest the Agile Principle above it within the software development process. The compiled list of Agile Practices (with references) are:

- *Embrace Change to Deliver Customer Value*
  1. Low process ceremony [60, 72]
  2. Client driven iterations [60]
  3. Continuous customer satisfaction feedback [64, 77]
  4. Evolutionary requirements [60]
  5. Reflect and tune process [64, 77]

- *Plan and Deliver Software Frequently*
  6. Agile project estimation [29]
  7. Smaller and more frequent releases (*4-8 weeks*) [64]
  8. Adaptive planning [60] [29]
  9. Risk driven iterations [60]
  10. Plan features, not tasks. [29]
  11. Maintaining a list of all features and their status (backlog) [57]
  12. Continuous delivery [60, 57, 45, 17]
  13. Planning at different levels [29]
  14. Collaborative planning [72, 27, 60]

- *Human-centric*

    15. Ideal agile physical setup [60]
    16. Self organizing teams [60, 72, 57, 27]
    17. Frequent face-to-face communication [72, 27, 18]
    18. Collaborative teams [80]
    19. Empowered and motivated teams [18]

- *Technical Excellence*

    20. Test driven development [16]
    21. Paired programming [84]
    22. No/minimal number of Cockburn's Level -1 or 1b people on team [24, 22]
    23. Daily progress tracking meetings [8]
    24. Agile documentation [73, 57]
    25. User stories [30]
    26. Continuous integration [60]
    27. Continuous improvement (refactoring) [57, 17, 41, 7].
    28. Unit tests [50]
    29. 30% of Cockburn's level 2 and Cockburn's level 3 people [24, 22]
    30. Software configuration management [57]
    31. Tracking iteration progress [60]
    32. No big design up front (BDUF) [5, 17]
    33. Coding standards [51, 82, 68]
    34. Knowledge-sharing tools [60]
    35. Task volunteering [60]

- *Customer Collaboration*

    36. Frequent (collocated) face-to-face interaction between developers & users [17]
    37. Customer immediately accessible [22]

38. Customer contract revolving around commitment of collaboration [45, 64]
39. Customer contract reflective of evolutionary development [45, 64]
40. Customer commitment to work with developing team [18]

This list is not prioritized or ordered in any way. The references associated with each practice serve as a good starting point for those who wish to gain further knowledge about the practice. A brief explanation of each practice is included in the next sections. After compiling the Agile Practices, the next step is to place each practice in its relative agile level, as illustrated by Figure 23. Each puzzle piece represents one or more Agile Practices.

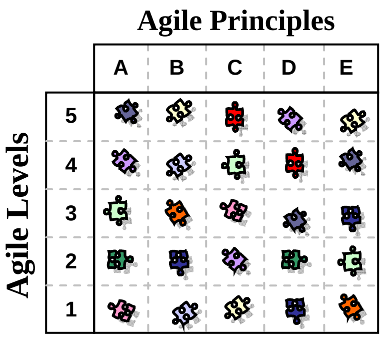


Figure 23. Agile Levels populated with Agile Practices categorized within Agile Principles

The next section focuses on the process by which each Agile Level is populated with Agile Practices, and how Agile Principles guide this process. The next section explains the population process through a detailed, working example of how Agile Level 1 is populated.

4.4.1 Populating Agile Level 1 with Agile Practices

To populate an Agile Level, one must first recognize the objective of the Agile Level. For example, the objective of Agile Level 1 is enhancing communication and collaboration in the software development process.

At this point, the Agile Level is empty and does not contain any Agile Practices (see Figure 24). The population process moves across the level, focusing on each Agile Principle separately. The previous section shows the Agile Practices classified under each Agile Principle. The approach taken to populate an Agile Level is to look first at each Agile Principle and the Agile Practices related to this principle and attempt to identify the Agile Practices that promote the overall objective of that Agile Level (in this case enhancing communication and collaboration).

	Agile Principles				
	Delivering Customer Value by Embracing Change	*Planning to Deliver Software Frequently*	*Human-centric*	*Technical Excellence*	*Customer Collaboration*
Level 1: **Collaborative**					

Figure 24. Agile Level 1 unpopulated with practices

Identifying Agile Practices related to the First Agile Principle

The first agile principle, delivering customer value by embracing change, provides a starting point for explaining the process of populating agile level 1. At this point, it is important to recall the objective of agile level 1, enhancing communication and collaboration, and then look at the set of Agile Practices related to the Agile Principle and ask which of the practices, when adopted, would enhance communication and collaboration within the development process. Below are all the Agile Practices that manifest the first Agile Principle:

- Low process ceremony
- Client driven iterations
- Continuous customer satisfaction feedback
- Evolutionary requirements
- Reflect and tune process

The key concern is to ascertain if these five agile practices, when adopted, will enhance communication and collaboration. Different people may have different answers, which is acceptable. This concept of accepting different answers is further discussed in Section 4.6, when the tailorability of the measurement index is discussed. With SAMI, the *"reflect and tune"* agile practice is included in the first level of agility, because this practice is concerned with holding retrospectives at regular intervals within the development process. The objective of this practice is also to tune the future use of the process to overcome any obstacles or challenges faced thus far.

Holding these reflecting and tuning retrospectives enhances communication and collaboration because they provide a forum where the stakeholders can express any process challenges they have encountered and suggest solutions for them for the next period of time. At the same time, this agile practice helps deliver customer value, because the stakeholders can discuss necessary changes in the process and actually embrace them. Ester Derby and Diana Larsen wrote an informative book about agile retrospectives [34].

	Agile Principles				
	Delivering Customer Value by Embracing Change	*Planning to Deliver Software Frequently*	*Human-centric*	*Technical Excellence*	*Customer Collaboration*
Level 1: **Collaborative**	Reflect and tune process				

Figure 25. Agile Level 1 populated with only one Agile Practice

At this point in the population process, one agile practice has been identified under the first agile principle (see Figure 25). If no other practices exist under that principle that help enhance communication and collaboration, then the population process moves to the second principle to identify any agile practices under the second principle

Identifying Agile Practices related to the Second Agile Principle

Moving along to the second agile principle of planning and delivering software frequently, within this level of agility continues the process of populating Agile Level 1. Again, it is necessary to look at all Agile Practices under the named principle and try to pick those practices that, once adopted, would enhance communication and collaboration. The next choice, therefore, is *collaborative planning*.

Collaborative planning is a common concept used in agile and other collaborative environments. The agile process encourages all stakeholders in the project to come together during the planning phase or activities of the project. Even when the project is large and composed of many teams, it is recommended that representatives from each be included. This practice increases project visibility and buy-in from different groups of stakeholders. Involving more people builds more loyalty and acceptance of the plan under development and increases the motivation and empowerment of the individuals on the team [72, 27, 60]. Other than participation in the decision-making process, collaborative planning is a powerful tool for information sharing, negotiation and participation.

The rationale behind choosing this practice for Agile Level 1 is that collaborative planning represents a step toward establishing a collaborative environment within the organization. Without collaborative planning, employees may feel left out and, therefore, not experience ownership of the project. Another possible impediment to a successful transition to agility is managers who are out of touch with the reality of the software development effort, or the developers themselves, and, therefore, develop a project plan that is impractical and causes frustration within the team. All these are symptoms of a non-collaborative environment. The use of collaborative planning enhances communication and collaboration within the organization because the client, the developers, and the managers all work together to set the project plan.

Identifying Agile Practices related to the Third Agile Principle

The human-centric principle is another agile principle within the first level of agility. There are many aspects to the human-centric nature of an agile software process. After the achievement of all five levels of agility, the organization realizes the full human-centric nature of agile software processes. However, since populating the first level of agility is the focus here, it was only necessary to select those Agile Practices that help *establish* a collaborative environment from a human-centric perspective. Consequently, in this level of agility there are two human-centric Agile Practices, *collaborative teams* and *empowered and motivated teams*.

In her latest book [80], Jean Tabaka defines a collaborative team. Her definition is comprehensive and includes some of the Agile Practices and concepts that are introduced in higher levels of agility. However, Agile Level 1 adopts only a subset of the characteristics she has defined for collaborative teams. At this level, supervisors must empower and equip these teams with the authority to make decisions on their own. This authority helps motivate the team members, who must believe that they can solve any problem that team faces. The team members must cooperate with each other and with other teams. Although these concepts seem simple, they are often overlooked within software development efforts. To establish true agility in a software development process, a conscious effort must be made to introduce and maintain these values and characteristics within the team.

	Agile Principles				
	Delivering Customer Value by Embracing Change	*Planning to Deliver Software Frequently*	*Human-centric*	*Technical Excellence*	*Customer Collaboration*
Level 1: **Collaborative**	Reflect and tune process	Collaborative planning	Collaborative teams Empowered and motivated teams		

Figure 26. Agile Level 1 after populated three Agile Principles

After identifying the Agile Practices that serve the first three Agile Principles, Agile Level 1 encompasses four Agile Practices (see Figure 26).

Identifying Agile Practices related to the Fourth Agile Principle

The populating of Agile Level 1 continues with the technical excellence principle, for which three Agile Practices are chosen. The practices selected are *coding standards*, *knowledge sharing tools*, and *task volunteering*. All three of these practices promote enhancing communication and collaboration from a technical excellence perspective.

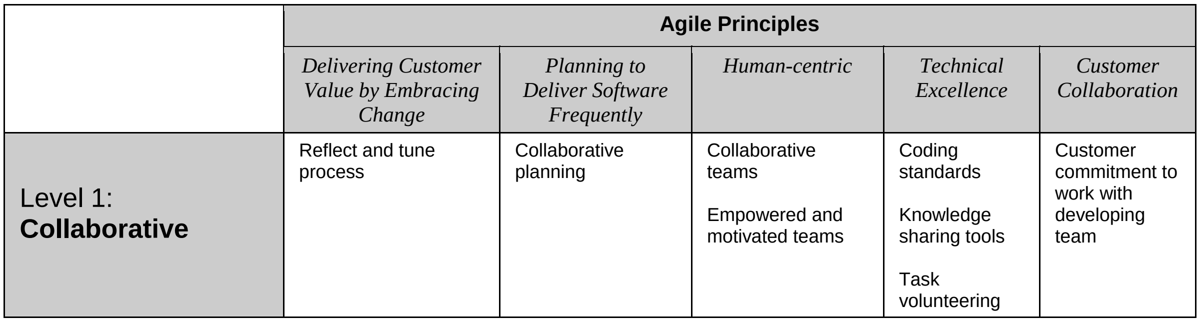
Even tools that focus primarily on promoting high quality software development have a communication or collaboration aspect to them. Coding standards are important for collaboration, because when people start collaborating and cooperating, coding standards help people understand shared code more easily [51, 82, 68]. The promoting of a collaborative and cooperative environment means that while the developers write code they need to make sure they can collaborate from a coding perspective. Coding standards facilitate that level of collaboration because they create a "common language" between all the developers. In other words, once someone opens the code of someone else, just by looking quickly through it and knowing these standards, he or she can read and comprehend the code with ease. It was interesting to find out that in the results of the March 2006 Agile Adoption Rate Survey conducted by Scott W. Ambler, have shown that the agile practice most often adopted by organizations was coding standards. About 40% of the more than 4000 participants said they had adopted common coding guidelines.

The second agile practice chosen for Agile Level 1 to enhance communication and collaboration from a technical perspective was the use of knowledge sharing tools. Knowledge sharing tools could be electronics such as wiki, blogs or they could be simple whiteboards or walls. Continuously sharing project information through these knowledge-sharing tools helps to establish a collaborative environment while at the same time sharing technical information and hence enhances the technical excellence of the overall software development process. Another benefit of having knowledge sharing tools is increasing project visibility, which, in turn, increases loyalty and the feeling of ownership. This helps support another practice discussed in the previous principle, empowered and motivated teams. In summary, knowledge sharing tools such as blogs, wikis and forums help to document and maintain the information and knowledge exchange that occurs between people and enables others to learn from that, thereby promoting even more collaboration [60].

The last practice chosen related to the technical excellence principle that promotes collaboration is to allow the developers to volunteer for tasks rather than have the manager assign the tasks to them. It is a rather simple practice that is carried out usually during planning meetings. In a planning meeting once the list of tasks has been generated, the project manager should encourage people to volunteer and commit to tasks that they choose based on their preference. Besides increasing the employee’s motivation and job satisfaction rate, the freedom that this practice gives results in a higher quality performance from the employees than they are likely to give when a manager assigns the tasks. If no one volunteers for a particular task, it is the team's collective responsibility to complete that task, instead of the managers responsibility to assign it to someone.

Identifying Agile Practices related to the Fifth Agile Principle

The last agile principle, "customer collaboration," suggests that to promote a collaborative environment the customer too has to commit to working with the development team. It is important to have "*customer commitment*" as a practice in Agile Level 1, because many Agile Practices found in upcoming Agile Levels depend on some degree of the customer's interaction and collaboration. Therefore, the customer needs to make a commitment at this level to work with the development team. Although some might think of the customer's willingness to collaborate as the default, in some situations a customer, reluctant to exert additional effort expects the contractor to provide the majority of the project's effort [57].



	Agile Principles				
	Delivering Customer Value by Embracing Change	*Planning to Deliver Software Frequently*	*Human-centric*	*Technical Excellence*	*Customer Collaboration*
Level 1: **Collaborative**	Reflect and tune process	Collaborative planning	Collaborative teams Empowered and motivated teams	Coding standards Knowledge sharing tools Task volunteering	Customer commitment to work with developing team

Figure 27. Agile Level 1 populated with Agile Practices

By identifying the Agile Practices under the last Agile Principle, the population process of Agile Level 1 is complete (see Figure 27). We emphasize that the placement of the practices in the SAMI is not the focus of the measurement index. The SAMI structure is what is fundamental. In addition, the positioning of the practices is subject to change from one user to another. We detail how this works in Section 4.6, which discusses the tailorability of the measurement index.

The procedure to populate Agile Levels 2 through 5 is similar to that of Agile Level 1. The following section briefly discusses Agile Levels 2 through 5 and their associated Agile Practices.

4.4.2 Agile Practices within Agile Level 2

Agile Level 2 encompasses a set of Agile Practices that work together synergistically to deliver software incrementally in shorter cycles. Similar to Agile Level 1, Agile Principles guide the process of populating Agile Level 2 with its corresponding Agile Practices. The following is a brief discussion about Level 2 Agile Practices populated from each Agile (Figure 28).

	Agile Principles				
	Delivering Customer Value by Embracing Change	*Planning to Deliver Software Frequently*	*Human-centric*	*Technical Excellence*	*Customer Collaboration*
Level 2: **Evolutionary**	Evolutionary requirements	Continuous delivery Planning at different levels		Software configuration management Tracking iteration progress No big design up front (BDUF)	Customer contract reflective of evolutionary development
Level 1: **Collaborative**					

Figure 28. Agile Level 2 populated with Agile Practices

Evolutionary Requirements is the most important agile practice that helps the delivery of software incrementally in shorter iterations. It suggests that requirements should be iteratively evolved instead of being fully developed in one major specification effort [60]. Agile practitioners argue that upfront requirements elicitation is ineffective. The reason is that requirements are bound to change. Therefore, eliciting known requirements and leaving the rest to evolve based on the customer’s feedback yields the best value to the customer and prepares the process to embrace change.

Continuous Delivery, another Agile Practice within Level 2, focuses on the notion of delivering the product in small iterations at regular intervals. The emphasis is on regular intervals, because having firm deadlines for each iteration ensures the team has to divide the product (and not deliver it all at once) in order to meet the given timeframe for each iteration. The duration of the iteration is irrelevant at this Agile Level. Another Agile Practice in Level 4 defines the duration of these iterations, and ensures they are short. On a side note, it is common to see agile practices in lower Agile levels to be of generic nature, and agile practices in higher Agile levels to be more specific. This is due to the fact that in the lower levels the project is still being introduced to these new agile concepts, and then later as they become more agile (i.e. aspire for higher agile levels) , these concepts will be manifested through more specific practices.

Since the development process is now delivering the product in iterations on a regular basis, another Agile Practice, *Planning at different levels,* is needed to ensure that the project teams maintain a plan for both the overall product development lifecycle and the individual iterations. Usually two levels of planning occur in agile projects: Release planning (dealing with the overall product) and iteration planning (dealing with current iteration).

From a technical excellence perspective, it is crucial now at this level to have some tool for *Software Configuration Management (SCM).* SCM tools help control the different versions and iterations of the software being developed. Another agile concept found in this level is *Tracking Iteration Progress,* which is concerned with the team having a means by which they can measure the progress of the development effort within each iteration. This concept does not dictate a particular method to fulfill this tracking. It emphasizes, however, the need to have it. In Agile Level 4, another Agile Practice (*Daily progress tracking meetings*) defines a particular way to achieve this agile concept.

No Big Design Up Front (No BDUF) is another agile practice that ensures the product is being developed using an evolutionary approach. BDUF is where a "big" design is created before coding and testing takes place, as in a typical waterfall development process. In Agile development design is not a one-time, upfront phase, it occurs throughout the development process.

The last Agile Practice within Agile Level 2 is *customer contract reflective of evolutionary development*. This practice ensures the customer understands the evolutionary nature of the development process. It also does not define individual milestones when all the requirements and design documents are completed. If the contract does so then the evolutionary nature of the whole process is in jeopardy. The contractor will want to meet these deadlines and therefore will not work using an evolutionary approach. Milestones in contacts that are reflective of the evolutionary development approach are usually defined around iterations or releases.

By adopting these agile practices the software development process can deliver software early and continuously and hence fulfills the goal of Agile Level 2. Once the development process is evolutionary (i.e. Agile Level 2 is archived), the project is ready for a higher level of agility. This is manifested in the project's ability to develop high quality working software in an efficient and effective manner which is the objective of Agile Level 3. The following section presents the different agile Practices in Agile Level 3 that help the level fulfill its objectives.

4.4.3 Agile Practices within Agile Level 3

Before achieving Agile Level 3, the organization should have already adopted all the practices in Agile Level 1 and 2. Now that communication and collaboration have been instilled in the development process, and the development process is delivering software early and frequently, the objective of Agile Level 3 is to increase the efficiency and effectiveness of the development process. This is achieved by encouraging the adoption of more engineering practices that enable the development of high quality working software. This is important because Agile Level 4 focuses on embracing change, and therefore Agile Level 3 must ensure the development process is effective, efficient, and stable (by adopting more engineering practices). To ensure the development process produces quality software in an efficient and effective manner, the SAMI suggests the adoption of nine different Agile Practices (see Figure 29).

Risk Driven Iterations help tackling risk elements as early as possible. Mitigating those risks early ensures that the project team does not spend a considerable amount of time building a system that they cannot complete. By catching these issues early, the development process becomes more effective.

	Agile Principles				
	Delivering Customer Value by Embracing Change	*Planning to Deliver Software Frequently*	*Human-centric*	*Technical Excellence*	*Customer Collaboration*
Level 3: **Effective**		Risk driven iterations Plan features not tasks. Maintain a list of all features and their status (backlog)	Self organizing teams Frequent face-to-face communication	Continuous integration Continuous improvement (refactoring) Unit tests 30% of level 2 and level 3 people	
Level 2: **Evolutionary**					
Level 1: **Collaborative**					

Figure 29. Agile Level 3 populated with Agile Practices

Another agile concept in Level 3 is to make sure that the team is *Planning Features not Tasks.* The customer usually expresses their needs as features in terms of the customer's domain vocabulary (e.g. an electronic shopping cart). Tasks, on the other hand, are usually expressed in the developer's terms and are not understandable to the customer (e.g. build an interface). One of the main reasons why planning should be done in terms of features not tasks is to prepare the development process for *Client Driven Iterations* (an Agile Practice in Agile Level 4). Another reason is that the planning is done in terms of features, when a certain feature changes or is removed the impact it has on the planning process is minimized.

Maintaining a list of all features and their status (i.e. backlog) is another agile practice in this level. The product backlog is a list of the business and technical requirements for the system being built or enhanced based on the current knowledge of the system. This practice includes the tasks for creating the backlog, and controlling it consistently during the process by adding, removing, specifying, updating, and prioritizing the backlog items. In this Agile Level the concern is not about who maintains this list or has authority to change it, the focus is to ensure such a backlog exists.

From a human-centric perspective, having an efficient and effective development process entails establishing efficient and effective communication between the members of the team. Having *frequent face-to-face communication* is one of the means to ensure that communication is more effective [24] [6]. Another human-centric Agile Practice is that of *self-organized teams.* A self-organized team is empowered by management to make decisions on their own without waiting for management's approval. These teams are usually cross-functional and the roles of individuals are not defined. When the team is given a task, it becomes the responsibility of the whole team, collectively, to finish it, not a specific person on the team. Management treats self-organizing teams as one entity without distinguishing between the individuals of the team. The importance of having self-organizing teams in this particular agile level is that, as the authors of the Agile Manifesto highlighted, "the best architectures, requirements, and designs emerge from self-organizing teams”, and it is in this Agile Level that it is crucial that the project develops the highest quality of working software.

In Agile Level 3, there are four agile practices found under the technical excellence agile principle. The first agile practice under this principle is *Continuous Integration*. Continuous Integration is a software development practice that encourages members of a team to integrate their work frequently. It is preferred that each integration is verified by an automated build tool in order to detect any integration errors as quickly as possible. This approach leads to significantly reduced integration problems and allows a team to develop cohesive software more rapidly [36]. Another Agile Practice recommended to improve effectiveness of the development process and make it ready to embrace change is that of *Continuous Improvement (aka Refactoring).* Refactoring is an essential practice to be adopted at Agile Level 3 because of the project evolutionary development process (assumed after Agile Level 2). Agile development addresses the issue of continuously developing and improving a system’s design by the use of refactoring [41] [7]. Refactoring involves rewriting the code to improve its structure, while explicitly preserving its behavior, and is sometimes informally referred to as "cleaning up the code". Therefore, regularly refactoring the code is necessary since evolutionary requirements are adopted (from Agile Level 2). In general, the refactoring process focuses on removing code duplication (a sign of poor design that might be introduced due to no Big Design Up Front). Additionally, refactoring increases the cohesion of the code, while lowering its coupling, which makes the system more ready to embrace change without “breaking” other parts of the system.

Refactoring is strongly supported by comprehensive testing to be sure that as the design evolves, nothing is broken. Thus, organizations are encouraged to adopt *Unit Tests,* another agile practice in this level. Unit tests are code procedures used to validate that individual units of source code are working properly. A unit of source code is the smallest testable part of an application. For example, in procedural programming a unit may be a function or procedure, while in object-oriented programming, the smallest unit is usually a class. A unit test provides a strict, written contract that the piece of code must satisfy. While in an agile development process it is highly recommended for unit testing to be automated through a testing framework, it may still be performed manually.

The last agile practice in Agile Level 3 is to have *30% of the team be Cockburn Level 2 or Cockburn Level 3 people.* Cockburn's people levels are directly related to the amount of experience the developer has. Inspired by the martial art of Aikido, Alistair Cockburn has discussed the three levels of listening at which people are placed [24]. Cockburn has identified three levels of understanding people have when they approach new material and has argued that a person's level of understanding is directly linked to his or her experience in the domain. Barry Boehm adapted these levels of generic understanding and created a measurement index to qualify the competency of the developer based on his or her understanding of the core concepts of programming, especially object oriented programming. While this measurement index is incomplete, it is a good starting point for creating a way to measure the competency of the developers not in terms of a set of generic skills, but in terms of their understanding of how to this approach object oriented programming. Table 11 identifies the characteristics expected from developers at each of the Cockburn's Levels.

Level	**Characteristics**
3	Able to revise a method (break its rules) to fit an unprecedented new situation
2	Able to tailor a method to fit a precedented new situation
1A	With training, able to perform discretionary method steps (e.g., sizing stories to fit increments, composing patterns, compound refactoring, complex COTS integration). With experience can become Level 2
1B	With training, able to perform procedural method steps (e.g. coding a simple method, simple refactoring, following coding standards and CM procedures, running tests). With experience can master some Level 1A skills.
-1	May have technical skills, but unable or unwilling to collaborate or follow shared methods.

Table 11. Cockburn's Levels

At Agile Level 3, the team needs to consist of developers who can handle new and unexpected problems because they are adopting many new technical practices, that is why Cockburn's Level 3 and Level 2 people are important elements of the team.

Before presenting the Agile Practices in Level 4, the is an important reason why the agile practice regarding the competency of the team members (Cockburn's Levels) is in the same Agile Level as that of self-organizing teams. The explanation stems from Ken Blanchard and Paul Hersey's theory of Situational Leadership, which explains the relation between the issue of personnel competence and self-organizing teams [44]. Blanchard and Hersey have characterized leadership style in terms of the amount of direction and support that the leader provides to his or her followers. They have categorized all leadership styles into four behavior types. Additionally their theory argues that leaders should not always use one leadership type, because leadership depends on two factors, the commitment and competence of the followers.

- *Directing Leaders:* make and announce decisions. Used with low competence, high commitment followers
- *Coaching Leaders:* seek ideas and suggestions from the followers, but make the decisions. Used with followers with some competence and low commitment
- *Supporting Leaders:* facilitate day-to-day decisions, such as task allocation and processes, that are now the responsibility of the followers. Used with followers that have high competence and variable commitment
- *Delegating Leaders:* give control to the followers and decide the time and extent of their own involvement. Used with high competence, high commitment followers

With this theory in mind, the agile practice of having *self-organizing teams* implies that management should give the employees control over the decision making process. This is similar to the Supporting and Delegating leadership styles, which

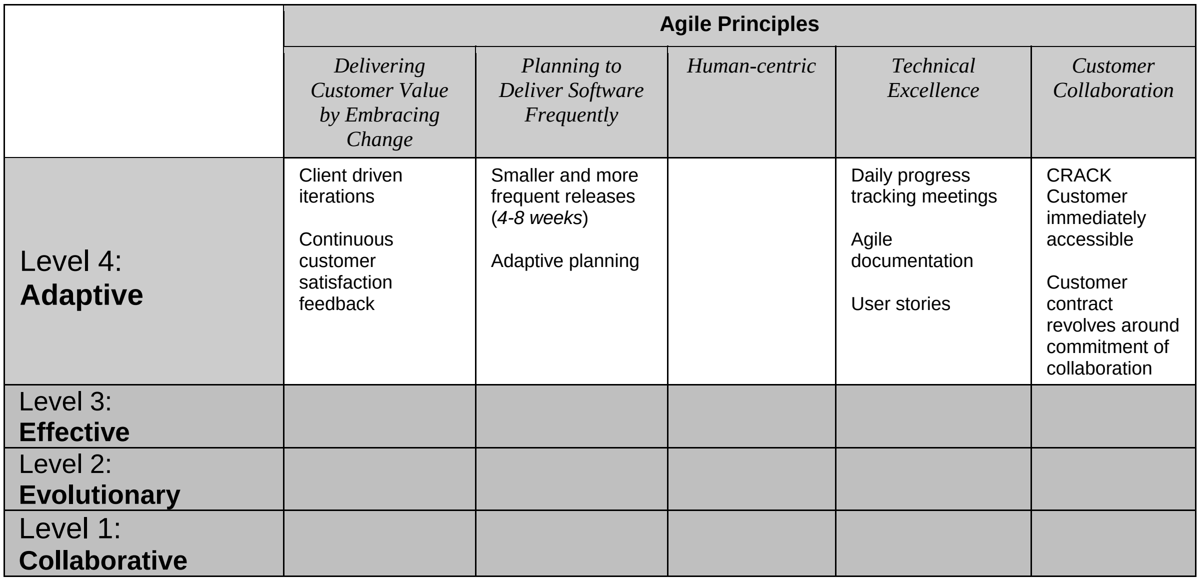
Blanchard and Hersey recommend should be practiced only when there are competent individuals on the team. This is why competence-related practices are not introduced until Agile Level 3, the same level in which the practice of *self-organizing teams* is introduced.

After adopting the Agile Practice in Agile level 1, 2 and 3, the development process should be ready to move to the next level of agility, Agile Level 4.

4.4.4 Agile Practices within Agile Level 4

The practices in Agile Level 3 focus on increasing the overall efficiency and effectiveness of the software development process by adopting certain engineering practices that help stabilize and automate the development process. Establishing these practices first helps prepare for Agile Level 4, which focuses on responding and embracing change through multiple levels of feedback. This section presents an overview of the nine agile practices that enable Agile Level 4 to achieve its objective (see Figure 30).

One of the key Agile Practices enabling the development process to be responsive to change (i.e., adaptive) is to have *Client Driven Iterations.* Client-driven iterations imply that the client dedicates the choice of features for the next iteration. This makes the client in control and able to change the system based on whatever they perceive as the highest business value to them. In this way, the client steers the project, iteration by iteration, requesting the features that they currently think are most valuable based on their latest insight, rather than speculatively at the start of the project. The customer has ongoing control to change the system as fresh information arises, and the development process should easily embraces these changes. Another agile practice that is supplementary to Client Driven Iterations is *Continuous customer satisfaction feedback.* Continuous feedback is crucial to ensure the customer is satisfied with what is being developed. If customer satisfaction or acceptance is only sought at the end of a project, then there is a significant risk that what has been developed is not what the customer actually needs. Moreover, when the customer requests a change in the system, it is crucial to gather their feedback to ensure the change implemented is satisfactory. If this practice is overlooked, more time and much effort are wasted. Hence in highly adaptive environments, such as Level 4 of agile software development, gathering frequent feedback from the customer is essential to ensure that the right product is being developed.



	Agile Principles				
	Delivering Customer Value by Embracing Change	*Planning to Deliver Software Frequently*	*Human-centric*	*Technical Excellence*	*Customer Collaboration*
Level 4: **Adaptive**	Client driven iterations Continuous customer satisfaction feedback	Smaller and more frequent releases (*4-8 weeks*) Adaptive planning		Daily progress tracking meetings Agile documentation User stories	CRACK Customer immediately accessible Customer contract revolves around commitment of collaboration
Level 3: **Effective**					
Level 2: **Evolutionary**					
Level 1: **Collaborative**					

Figure 30. Agile Level 4 populated with Agile Practices

Looking back at Agile Level 2, under the “Planning to deliver Software Frequently” Agile Principle, there is a practice named “Continuous Delivery.” A more specific version of this agile practice is what is found in Agile level 4, under the same principle, and is titled “*Smaller and more frequent releases (4-8 weeks)*.” Smaller and more frequent releases (4-8 weeks) encourages organizations to keep the timeframe for the releases small and limit them to 8 weeks. This practice focuses on the timeframe for each release not iteration. Usually building a release will include multiple iterations. Therefore, while not explicitly restricting the timeframe for iterations, naturally when the release timeframe is reduced, the iteration timeframe should be reduced accordingly. Having shorter releases helps the development process's ability to embrace change. Once a release is completed, feedback is obtained and used to adapt the next release. If the time between releases is long, it means additional time and effort will be invested into the product before feedback is obtained. The risk is that this feedback may change the direction of the product, and therefore it would have been much better to find that change of direction as quick as possible. The next, and closely related, agile practice is *Adaptive Planning.* In general*,* the more time spent on a project, the more knowledge one acquires about it. Many people invest so much time creating a plan for the entire project at the beginning (when they have the least knowledge about the project) and avoid planning during the project (when more information becomes apparent). Adaptive Planning delays developing an iteration details until immediately before the following iteration, and therefore incorporates all the feedback obtained including what is learned from the previous iteration. Adaptive planning helps organizations embrace change because the focus shifts from adapting to a plan (which make people less embracing of change) to continuously planning based on the latest feedback obtained (which inherently promotes the culture of welcoming change).

In Agile Level 4, under the technical excellence principle, there are three agile practices. The first is *daily progress tracking meetings.* As mentioned earlier, this agile practice is a more specific version of another agile practice (Tracking Iteration progress) located in Agile Level 2. The key to *daily progress tracking meetings* is the word daily. From the start of Agile Level 4 one can notice the fast-pace nature of this level. This is where the agility of the development process is really put to test. The development process will need to respond to growing demands while still producing high quality working software. Therefore, to survive in a fast-paced adaptive environment, the team must stay informed on a daily basis regarding the status of the iteration. Another important agile practice that is *Agile Documentation* [73]. Until Agile Level 4, documentation is not mentioned, despite the fact that it is one of the first "negative" things mentioned about Agile Software development.

Nevertheless, Agile Documentation discourages writing unnecessary requirements, design or management documents for two main reasons [17] [60]. The first reason is that agile practitioners have recognized that there is a cost associated with developing and maintaining documentation. The second reason is that documentation does not deliver value to the customer, while working software does. Therefore, minimal documentation is promoted in agile development, because it gives the customer a better and earlier return on investment – valuable working software instead of perceived invaluable documents.

The last Agile Practice under the technical excellence agile principle is *User Stories.* A user story is an agile practice where software system requirements are formulated as one or two sentences in the everyday language of the user. User stories, together with acceptance tests, are used for the specification of requirements in some agile software engineering methods. Each story must be short enough to fit on the side of a note card. The customers should write the stories for the software project; these stories are the way they influence development. User stories are, therefore, a quick way of handling customer requirements without having to deal with large formal requirement documents and tedious tasks related to maintaining them. This helps the development process respond faster and with less overhead to rapidly changing real-world requirements.

The final two Agile Practices within Agile level 4 are related to Customer Collaboration. The first practice is to have the *CRACK Customer immediately accessible.* A CRACK Customer refers to a customer who is *C*ollaborative, *R*epresentative, *A*uthorized, *C*ommitted and *K*nowledgeable. These characteristics are important because they address a number of issues that emerge during the development process. At this level, the focus is not where this CRACK customer is located, the focus is on how responsive they are. To cope with the fast-paced environment where change is welcomed and embraced, the team must have immediate access to a CRACK Customer to address all their questions and concerns as they arise. The next, and last, Agile Practice in this level is to have the c*ustomer* *contact revolve around commitment of collaboration* not features or requirements. This is one of the ultimate factors that enables organizations to embrace change and actually welcome it. Usually the customer contract contains a list of features or requirements that one must deliver or else they are at risk of not getting the money or legal prosecution. In either way the organization usually prefers to "not welcome change" and just deliver what they were contracted for and get the money. What this practice want to achieve is to have the customer agrees to contract the degree and amount of collaboration we have not the list of requirements or features.

With the adoption of all the practices in Agile Level 4, the organization aspires for even a higher level of agility, Agile Level 5. The next section presents the agile practices found within Agile Level 5.

4.4.5 Agile Practices within Agile Level 5

Agile Level 4 builds upon the communication and technical foundations established by Agile levels 1 through 3. Agile Level 4 takes the development process to a new level of agility by enabling it to respond to change. The next level of agility, Agile Level 5, focuses not only on the development process, but also on the culture. Agility is essentially a culture, and it is important to have an environment that is reflective and supportive of the agile nature of the software development process. Therefore, Agile Level 5 concentrates on establishing an all-encompassing environment to sustain and foster agility throughout the organization. This section presents the seven agile practices that enable Agile Level 5 to achieve its objective. These can be seen in Figure 31.

The first agile concept is that of having *Low Process Ceremony* in the organization. Process Ceremony is the level of paperwork involved with a process. For example, an organization with high process ceremony would require that change requests be signed by at least three levels of management. It is obvious that high process ceremony jeopardizes the very essence of agility, being responsive to change.

	Agile Principles				
	Delivering Customer Value by Embracing Change	*Planning to Deliver Software Frequently*	*Human-centric*	*Technical Excellence*	*Customer Collaboration*
Level 5: **Encompassing**	Low process ceremony	Agile project estimation	Ideal agile physical setup	Test driven development Paired programming No/minimal number of level -1 or 1b people on team	Frequent face-to-face interaction between developers & users (collocated)
Level 4: **Adaptive**					
Level 3: **Effective**					
Level 2: **Evolutionary**					
Level 1: **Collaborative**					

Figure 31. Agile Level 5 populated with Agile Practices

Under the "Planning and Deliver Software Frequently" agile principle, most of the practices are related to planning. However, in Agile Level 5 the focus moves to a more fundamental point, which is *Agile Project Estimation*. This is an important practice because plans are only as good as the estimates they are based on. At the same time, highly dynamic and changing environments, such as the agile environment, needs an estimating technique that is flexible without jeopardizing accuracy. Agile Project estimation can be done in various ways; two of the most popular are story points and ideal days [29].

From a human-centric perceptive, having the *Ideal agile physical setup* helps establish the right environment for the agile software development to thrive in. Many different physical setups are suitable for agile development. The key to most of these physical setups is to have the whole team together in one room with no cubicles or dividers. The objective is to have as much tacit knowledge flow between all the team members. Accessibility to team members becomes instantaneous and lines are immediately established. The physical setup of an agile working environment also deals with the way the tables are setup in the room, the equipment available in the room (white boards, projectors…etc) and also the way the walls are utilized as information radiators [24] [45].

A famous agile practice is *Test Driven Development (TDD)*. Test-Driven Development (TDD) is a software development technique that involves repeatedly first writing a test case and then implementing only the code necessary to pass the test. Practitioners emphasize that test-driven development is a method of designing software, not merely a method of testing. Adopting TDD results in a culture change related to testing, designing and requirements. Additionally, adopting TDD requires that the developers have a certain level of comfort with automated unit testing. There is a rhythm to TDD. First the developer creates one test to define some small aspect of the problem at hand. Then s/he creates the simplest code that will make that test pass. Then s/he creates a second test. Now the programmer adds to the code they just created to make this new test pass, but no more! Not until they have yet a third test. S/he continues until there is nothing left to test. The code created is usually simple and concise, implementing only the features wanted. Other developers can see how to use this new code by browsing the tests.

Another Agile Practice that is reflective of a culture change is *Paired Programming*. Paired programming is an agile practice that requires two software engineers to participate in a combined development effort at one workstation. Each member performs the action the other is not currently doing; for example, while one types in unit-tests the other thinks about the class that will satisfy the test. The person doing the typing is known as the driver, while the person guiding is known as the navigator. It is often suggested that the two partners switch roles at least every half-hour or after making a unit test. When an organization starts using Paired

Programming an environment of technical excellence is created because of the immense sharing of expertise that occurs from developers working together. Additionally, a sense of collective ownership of the code emerges where code is not owned by one person, but multiple people.

Since the competence levels of developers plays a significant part in agile software development, the SAMI requires the team to have more competent people on it at the higher levels of agility. A project aspiring for Agile Level 5 should have *no or a minimal number of Cockburn level -1 or 1b* people on the team. As explained earlier in 4.4.3 Cockburn Level 1B in Level -1 developers have the least experience and are not able or willing to collaborate which can hamper the transition to agility; this is why their presence is discouraged at Agile Level 5. Agile practitioners have said that competent people are necessary because of the nature of the fast-paced development process and the constant need to be responsive to change.

The final agile practice in Agile level 5 is to have *frequent face-to-face interaction between developers and users (collocated).* It is ideal to have not just the developers collocated by also have the customer and users in the same room. This ensures almost instant feedback and incredible communication.

The previous explanations about the positioning of the Agile Practices show that there is synergy created when the practices of each agile level are put together and adopted together. That is why it is not recommended to encourage people to jump levels and try to adopt practices of a higher level before they have adopted all the practices in the lower levels. Table 12 consolidates all the Agile Levels and their Agile Practices, categorized by the Agile Principle they fall under. It is important to emphasize that the population of the SAMI is not fixed, and is subject to people's experience.

In the previous sections we highlighted our rational for positioning each practice in their respective levels. Through time and experience our opinions may change on where certain practices would be best situated. What is key to remember is that the positions of the practices in the measurement index is not the focus of the measurement index, it is the structure of the SAMI that is fundamental. The tailorability of the SAMI is discussed in greater detail in Section 4.6

	Agile Principles				
	Embrace Change to Deliver Customer Value	*Plan and Deliver Software Frequently*	*Human-centric*	*Technical Excellence*	*Customer Collaboration*
Level 5 **Encompassing** *Establishing a vibrant environment to sustain agility*	Low process ceremony [60, 72]	Agile project estimation [29]	Ideal agile physical setup [60]	Test driven development [16] Paired programming [84] No/minimal number of level -1 or 1b people on team [24, 22]	Frequent face-to-face interaction between developers & users (collocated) [17]
Level 4 **Adaptive** *Responding to change through multiple levels of feedback*	Client driven iterations [60] Continuous customer satisfaction feedback [64, 77]	Smaller and more frequent releases (*4-8 weeks*) [64] Adaptive planning [60] [29]		Daily progress tracking meetings [8] Agile documentation [73, 57] User stories [30]	CRACK Customer immediately accessible [22] Customer contract revolves around commitment of collaboration [45, 64]
Level 3: **Effective** *Developing high quality, working software in an efficient an effective manner*		Risk driven iterations [60] Plan features not tasks. [29] Maintain a list of all features and their status (backlog) [57]	Self organizing teams [60, 72, 57, 27] Frequent face-to-face communication [72, 27, 18]	Continuous integration [60] Continuous improvement (refactoring) [57, 17, 41, 7]. Unit tests [50] 30% of level 2 and level 3 people [24, 22]	
Level 2: **Evolutionary** *Delivering software early and continuously*	Evolutionary requirements [60]	Continuous delivery [60, 57, 45, 17] Planning at different levels [29]		Software configuration management [57] Tracking iteration progress [60] No big design up front (BDUF) [5, 17]	Customer contract reflective of evolutionary development [45, 64]
Level 1: **Collaborative** *Enhancing communication and collaboration*	Reflect and tune process [64, 77]	Collaborative planning [72, 27, 60]	Collaborative teams [80] Empowered and motivated teams [18]	Coding standards [51, 82, 68] Knowledge sharing tools [60] Task volunteering [60]	Customer commitment to work with developing team [18]

Table 12. SAMI populated with Agile Practices

The identification of the agile level and its principles and the Agile Practices and concepts inside each level, has completed the foundation of the agile measurement index. The last component, the indicators, is the actual means used to measure the degree of agility for a development process. The next section discusses indicators in detail.

4.5 Indicators

Thus far, SAMI, with its levels, principles and practices, only offers a roadmap for agile adoption efforts. Its levels and practices give organizations guidance about which practices to adopt first and the sequence of adoption. Moreover, the measurement index, as detailed thus far, is not capable of assessing the agile potential of a project or organization, because it lacks assessment indicators.

4.5.1 Introduction to Indictors

Basically, indicators are questions the assessor uses to assess certain characteristics of an organization or project, such as its people, culture and environment, in order to determine the entity's current or target level of agility. A set of indicators, or questions, should accompany each agile practice or concept in the measurement index. These indicators are used to assess the readiness of the organization or project to adopt an agile practice The SAMI employs over 300 indicators developed for the Agile Practices identified within the agile measurement index.

The objective of the indicators is to assess how ready the organization is to adopt a certain agile practice. Therefore, the indicators associated with each agile practice are concerned with assessing organizational characteristics that could influence the extent to which the organization is ready to adopt that agile practice. The organizational or project characteristics usually assessed to determine the readiness for a practice are:

- *Customers:* the project's customers and clients
- *Developers*: the technical staff involved with the development of the project
- *Managers*: the managers or executives overseeing the project and involved with decision making
- *Tools*: the software tools used within the organization or for a certain project
- *Culture*: the overall culture of the people within an organization or the project team
- *Project management*: the procedures and practices related to managing projects in the organization
- *Software Process*: the activities and artifacts related to the software development process in the organization
- *Physical environment*: the physical layout of the organization and the geographical and geo-spatial distribution of its employees

Since indicators are essentially questions, someone must be responsible for answering them. The customer is responsible for answering some of the indicators, while the developers and the managers answer still another group of indicators. Each of the indicators attempt to assess the group's opinion or feeling toward a certain concept or idea. The assessor usually answers indicators related to observing an activity or reviewing an artifact. Table 13 and Table 14 list some sample indicators that are used to assess the extent to which an organization is ready to adopt the agile practice of *collaborative planning*. Table 13 depicts some sample indicators that the developers should answer, and Table 14 shows some sample indicators that the managers should answer.

Indicator #	Statements	Nominal Values				
OR1_D1	Your manager listens to your opinions regarding technical issues	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR1_D5	You would like to participate in the planning process of the project you will work on.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR1_D7	It is acceptable for you to express disagreement with your manager(s) without fearing their retribution.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree

Table 13. Sample indicators to be answered by the developer

Indicator #	Statements	Nominal Values				
OR1_M3	You usually seek your subordinates' opinions before making a decision.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR1_M5	You frequently encourage your subordinates to find creative solutions to problems.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR1_M9	Developers should aid in the planning of a project.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree

Table 14. Sample indicators to be answered by the manager

The indicator number (first column) is used to identify and reference the indicator. The first two letters in the indicator's number (OR) mean that these indicators are concerned with assessing organizational readiness. The next numeral (1) represents the agile level of the agile practice to which these indicators are related. Since collaborative planning is an agile practice within Agile Level 1, the numeral used is (1). The letter after the underscore (_) refers to the type of person who should provide an answer for the indicator. The four letters used for that position are:

- A: denoting an indicator that the assessor, or the person conducting the assessment, needs to answer
- D: denoting an indicator that the developer, or anyone on the development side of the project, needs to answer
- M: denoting an indicator that a manager, or anyone performing management related tasks for to the project, needs to answer

- *C*: denoting an indicator that a customer executive, or a decision maker from the entity contracting to develop the software, needs to answer

A sequential number is used as the last digit in the indicator's number. The indicator itself (which can be seen in the second column) is usually a statement or question that needs a response. Most indicators are based on a 5-point Likert summated scale (third column), from 1 "strongly disagree" to 5 "strongly agree." A small number of indicators are based on other 5-point scales that are more appropriate to the organizational characteristic being assessed.

4.5.2 Organization of the Indicators

This discussion of the sample indicators provides a starting point for an explanation of the organization and structure of these indicators within the agile measurement index. Each agile practice is associated with a set of indicators that measure the extent to which the organization is ready to adopt the associated agile practice. The Goal Question Metric approach (GQM) [13] and the Objectives Principles Attributes (OPA) Framework [9] influenced the approach used to devise the indicators for each practice. Each indicator is designed to measure a particular organizational characteristic (discussed earlier) necessary for the successful adoption of the agile practice associated with the indicator. In other words, the goal is to assess the state of a certain organizational characteristic and then determine if the state of that characteristic is supportive of the adoption of the practice. For example, to assess the readiness of an organization to adopt collaborative planning, the assessor need to ascertain if its management style is suitable for the adoption of the agile practice. Hence, *Management Style* is the organizational characteristic assessed. The goal behind assessing the management style is to determine whether a collaborative or a command-control relation exists between managers and subordinates. The management style is an indication of whether management trusts the developers and vice-versa.

Therefore, the first step in identifying the indicators for an agile practice is to identify the organizational characteristics that need to be assessed in order to determine the extent to which the organization is ready to adopt the practice. Figure 32 shows the different organizational and project characteristics that need to be assessed for the Agile Practices in Level 1.

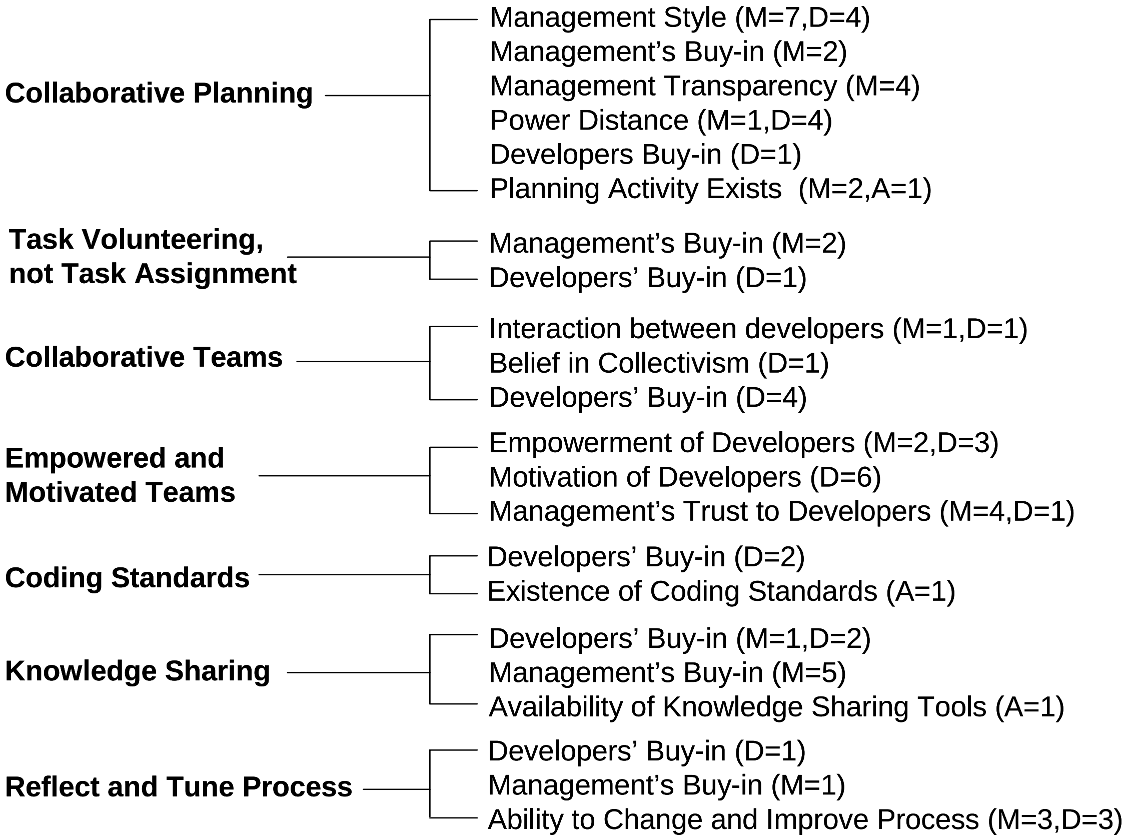

Collaborative Planning
Management Style (M=7,D=4)
Management's Buy-in (M=2)
Management Transparency (M=4)
Power Distance (M=1,D=4)
Developers Buy-in (D=1)
Planning Activity Exists (M=2,A=1)
Task Volunteering, not Task Assignment
Management's Buy-in (M=2)
Developers' Buy-in (D=1)
Collaborative Teams
Interaction between developers (M=1,D=1)
Belief in Collectivism (D=1)
Developers' Buy-in (D=4)
Empowered and Motivated Teams
Empowerment of Developers (M=2,D=3)
Motivation of Developers (D=6)
Management's Trust to Developers (M=4,D=1)
Coding Standards
Developers' Buy-in (D=2)
Existence of Coding Standards (A=1)
Knowledge Sharing
Developers' Buy-in (M=1,D=2)
Management's Buy-in (M=5)
Availability of Knowledge Sharing Tools (A=1)
Reflect and Tune Process
Developers' Buy-in (D=1)
Management's Buy-in (M=1)
Ability to Change and Improve Process (M=3,D=3)

Figure 32. Organizational Characteristics assessed for practice in Agile Level 1

While the typical structure for an indicator hierarchy is a tree (as in Figure 32), there are cases when the indicator structure can be an acyclic graph. This occurs when the same question (indicator) is used to assess two different organizational characteristics or when two different agile practices depend on the same organizational characteristic. This distinction between the structure being a tree or an acyclic graph has no direct impact on the assessment process but it is a point that is worth mentioning.

The assessor has two ways to identify the organizational characteristics that need assessment to determine an organization’s readiness for an agile practice. The first is commonsense and the second is experience and technical literature. The first is self-explanatory, because determining the organizational characteristics for some Agile Practices is a matter of common sense. For example, making sure that there is a developers’ buy-in before adopting coding standards is common sense. However, the main source of inspiration and the basis for selecting organizational characteristics to be assessed to determine the organizational readiness to adopt a practice consists of information gathered from agile consultants and coaches and the technical literature. Both of these sources are based on consultants’ experiences. Besides using common sense for some obvious Agile Practices, technical literature is the second source for identifying organizational characteristics used in assessing the readiness of an agile practice in SAMI. Some of these sources include [71, 57, 67, 80, 19, 31, 76].

Category of Assessment	Area to be assessed	Characteristic(s) to be assessed	To determine:	Assessment Method	Sample Indicators
People	Management	Management Style	Whether or not a collaborative or a command-control relation exists between managers and subordinates. The management style is an indication of whether or not management trusts the developers and vice-versa.	Interviewing	OR1_M1, OR1_M2, OR1_M3, OR1_M4, OR1_M5, OR1_M14, OR1_M17, OR1_D1 OR1_D2, OR1_D3, OR1_D4,
		Buy-In	Whether or not management is supportive of or resistive to having a collaborative environment	Interviewing	OR1_M9, OR1_M10,
		Transparency	Whether or not management can be open with customers and developers – No politics and secrets	Interviewing	OR1_M6, OR1_M7, OR1_M8, OR1_M13
	Developers	Power Distance	Whether or not people are intimidated/afraid to give honest feedback and participation in the presence of their managers	Interviewing	OR1_M11, OR1_D6, OR1_D7, OR1_D8, OR1_D9
		Buy-In	Whether or not the developers are willing to plan in a collaborative environment	Interviewing	OR1_D5
Project Management	Planning	Existence	Whether or not the organization does basic planning for its projects	Observation	OR1_A1
				Interviewing	OR1_M16, OR1_M18

Table 15. Readiness Assessment Table (RAT) for Collaborative Planning

Once the assessor has identified the appropriate organizational characteristics, they are then placed in a Readiness Assessment Table (RAT). Table 15 shows the RAT used to assess an organization's readiness to adopt collaborative planning.

The first column in the RAT highlights the generic organizational area to be assessed. The next two vertical columns specify the aspect of that area needing assessment. The third column designates the actual organizational characteristic to be assessed. The goal behind the assessment of this characteristic is defined in the fourth column titled "to determine." The fifth column provides information on the method used to conduct the assessment. The last column provides a reference to the actual indicators (presented earlier) that will be used to assess each particular organizational characteristic.

Appendix A provides the RATs for each agile practice along with all the indicators used to assess the Agile Practices in SAMI. This appendix also provides a detailed explanation on how to aggregate the different indicators together and calculate a final nominal value for each organizational characteristic. The next chapter provides information on how to interpret the results of the assessments.

As explained earlier in Section 3.2, the SAMI is tailorable. The five Agile Levels presented earlier are one instance of the SAMI. The next section provides a more detailed discussion of the tailorability of the SAMI.

4.6 Tailorability of the 5 Levels of Agility

Although the practices are arranged in a certain way within each agile level, it is understandable that others might have different points of view about the placement of certain Agile Practices. That is acceptable and does not jeopardize the structure of the agile measurement index. It merely means that it is possible to have more than one instance of the measurement index. The science of agile measurement indexes is still in its infancy and much more refinement is needed to consolidate or establish relations between the different instances of the Sidky Agile Measurement Index.

When members of the agile community viewed the five Levels of Agility, along with all their practices and indicators, several of its leaders encouraged the consideration of factors that might lead to alternate instances of the five Levels of Agility. These factors are *incorporating business values* and *reorganizing the practices based on experiential success*. The two following subsections elaborate on these factors and the tailorability of the five Agile Levels

4.6.1 Incorporating Business Values

Business values refer to the added benefit an organization realizes after adopting Agile Practices. For most organizations, the achievement of these business values is the real incentive behind adopting agility. For example, common business values that organizations hope to realize from adopting Agile Practices comprise decreasing time to market or increasing product quality. Augustine [75] and Elssamadisy [38] have suggested the possibility of prioritizing the levels of agility according to the business values an organization hopes to realize. This suggestion is both valuable and beneficial to the growth the framework, because currently the five Levels of Agility are not associated with any business values; instead they are based on the qualities and values of agility. This relationship between agility and business values is parallel to that between the *Agile Manifesto* (focusing on agile values) and the *Declaration of Interdependence* (capturing the business values) [2] [1].

4.6.2 Reorganizing Practices based on Experiential Success.

The agile coaches and consultants Cockburn [23], Cohn [28], and Wake [81], in addition to others, have suggested a reorganization of the Agile Practices based on experiential successes. That is, they advocate that the kind of projects and the experiences gained from previous adoption efforts can, and should, serve as a basis for formulating a better arrangement of the practices within the Agile Levels. For example, Cohn has suggested that *user stories* be introduced in the first level of agility, because, from his experience, they enhance collaboration and communication between the stakeholders with regard to requirements. Others have suggested that *pair programming* be included in the first level, because it helps to establish collaboration within teams. This inability to reach a consensus on the position of Agile Practices emphasizes an important factor in providing guidance in an agile adoption effort: the *adherence* to Agile Principles, not the positions of the actual practices, is paramount when establishing the levels. The intention behind the levels of agility is to provide a framework to guide the adoption process, not to dictate it.

The above rationalizations have lead to the conclusion that a tailorable measurement index is both desirable and beneficial. However, when tailoring or creating another instance of an agile measurement index, it is important to observe the following guidelines to ensure that the new measurement index has all the necessary components and a valid structure:

- *Ensure that multiple levels exist*. Levels are needed to enumerate the degrees of agility. Without levels, the power of the measurement index, when used in conducting comparative measurements of the agility, is diminished.
- *The measurement index is based on practices and concepts*. Agile Practices and concepts form the foundation of the agile measurement index. The extent to which Agile Practices and concepts can be adopted determines the agility of a process.
- *Each practice or concept has indicators*. When introducing a new agile practice (other than the 40 identified) to the measurement index, it is important that the practice has an associated set of valid and sufficient indicators. Without indicators, there are no means for conducting this assessment.

After the Agile Adoption Framework was developed, members of the agile community were asked to attend presentations about the framework and provide their feedback. The next chapter provides a detailed analysis of the results gathered from the substantiation process of the Agile Adoption Framework by the agile community.

5. Substantiating the Agile Adoption Framework

This chapter presents data that substantiates the validity of the Agile Adoption Framework. It describes the substantiation process and the rationale behind it. It also presents and analyzes formal and informal feedback collected from interviews, surveys and informal conversations from members of the agile community.

A longitudinal study is a common method for validating any new research framework. In this kind of study, one or more organizations use a new framework, and a researcher, or team of researchers, gather data that allows them to evaluate the process. Therefore, a longitudinal study is the ideal way to validate the Agile Adoption Framework. This is because observing and gathering data about the adoption process and comparing it to adoption processes at organizations not using the framework generates the empirical evidence needed to substantiate the validity of the framework.

However, the problem with this kind of study is that it takes extensive periods of time and requires several organizations to buy into the use of Agile Adoption Framework before enough empirical evidence can be gathered. The latter requirement intensifies the problem because many organizations are hesitant to employ a new framework with no evidence of its validity. Therefore, instead of a longitudinal study, gathering feedback from the agile community is the best means of substantiating the validity (or goodness) of the framework. This feedback should highlight the positive impact and benefits of the Agile Adoption Framework, as well as identify any potential problems. Moreover, substantiating the framework with feedback from the agile community can enable a future longitudinal study, because this type of positive feedback can increase the credibility of the framework in the eyes of the organizations, thereby increasing their willingness to participate in such a study.

The next section describes the approach used to gather feedback from the agile community.

5.1. The Substantiation Approach

Much thought and discussion was needed to identify an efficient and practical approach of gathering feedback from people in the agile industry, and who have little time in their schedules to invest in such studies. The following sections present the procedures used to gather feedback as well as a discussion about the participants and the surveys used.

5.1.1 Procedure for gathering feedback

The initial idea for gathering agile industry feedback was to prepare a comprehensive survey of the Agile Adoption Framework and to mail it to people in the agile community. However, since these potential participants were not yet informed about the framework, this survey had to be supplemented with reading material explaining the Agile Adoption Framework. Compiling an explanation of the framework to accompany the survey resulted in a 50-page document. The reality is that people in agile community, especially industry leaders and experts, would not have the time to read a 50-page document and fill out and return a survey.

Another idea, to visit and present the framework face-to-face to selected clients from the agile community, grew out of the spirit of agility and its human-centric nature. Visiting people in person ensured having their attention for at least the duration of the visit. On average, those who agreed to participate dedicated 90 minutes of their time for a visit. The next challenge was to develop an approach for presenting the framework and gathering feedback within the allocated time. The 12-

page survey, (included in the Survey Collection document) originally designed to gather feedback about all aspects of the framework was not suitable for this new approach. In order to assure a response at the time of the visit, the survey had to be short and concise. Otherwise, the participants would ask to fill it out and send it in later, and might become too busy to find the time to do so. Therefore, again employing the principles of agility, the questions providing the most valuable answers for the substantiation process were chosen. The result was a two-page survey in which each page provided a series of questions designed to elicit a response to each of the two components of the framework: the five levels of agility and four stage process. Analysis of the answers to the survey provided an overview of the agile community's response to the Agile Adoption Framework.

On the day of the presentation, each participant was given the two-page survey and a printout of the slides, including a short narrative for each slide. The participants generated interesting discussions because they were intrigued by the idea of the framework and the Sidky Agile Measurement Index (SAMI). These discussions served as the informal feedback used as part of the substantiation process. After the presentation and the discussions, the participants filled out the survey. Although discussions and surveys completed at a time of presentation guaranteed adequate feedback to validate the agile framework, the participants were given the option of completing the 12-page survey. Those interested in participating in a more elaborate feedback process were given the detailed assessment package (the original 50 page document) along with a self-addressed envelope to use to return the 12-page survey. There was enough enthusiastic interest in providing further feedback about the framework that over 65% requested the detailed assessment package. However, as anticipated, only a small fraction of those actually sent back the long survey.

The next section presents the structure and content of the surveys and questions used to gather feedback from the participants.

5.1.2 Overview of Survey Questions and Participants

Since the Agile Adoption Framework consists of two components, the SAMI and the 4-Stage process, feedback about each of the components was gathered separately for more accurate results. In the two-page survey, one page served for collecting data on the SAMI and the other for collecting data on the 4-Stage process. The advantage of using what amounted to separate one-page surveys is to enable the competitive analysis of the results between the two components of the framework. Since the two-page surveys were intended for managers and consultants, whose time and availability is limited, there were 20 quantitative questions based on a 5-point Likert summated scale, from 1 "strongly disagree" to 5 "strongly agree,". There were also 4 qualitative open-ended questions that complemented the quantitative questions to provide broader and deeper coverage of the topic. These qualitative questions, coupled with the informal discussions, helped to elicit better feedback from the participants about the framework. Thirty-five members of the agile community agreed to participate in the survey process. Of these, 27 filled out the two-page survey after the 60 minutes presentation of the Agile Adoption Framework. Of the five that asked to complete the survey and mail it in, only one did so. This confirms the importance of having participants fill out the survey at the time of the presentation. The 28 completed surveys yielded a 78% response rate.

The 12-page survey was offered to each of the 35 participants, but only 12 took it to fill out and return by mail. Of these, only two completed and returned them. These two surveys represent 17% of the 12 participants and, including these two participants, 7% of the 28 who filled out the two-page survey. While this response was disappointing, it was expected. Anonymous copies of the 28 two-page surveys and the two 12-page surveys can be found in Appendix B.

Each of the next two subsections presents and discusses the questions used to gather feedback on each of the components of the framework.

Questions about the Sidky Agile Measurement Index

Since it would have been difficult to gather feedback about the five levels of agility, along with their principles, practices and indicators within an acceptable timeframe, a two-page survey was designed to optimize time and response data. On the page devoted to the SAMI, the five questions measured, using the Likert scale, the comprehensiveness (using 2 questions), practicality, necessity and relevance of the location of the practices within the SAMI (see Table 16).

LIKERT – SCALED QUESTIONS	
TO DETERMINE …	QUESTION(S) USED
Comprehensiveness	The 5 agile levels are comprehensive enough to depict the various states and levels most organizations could be at relative to agility?
	The 5 agile levels are defined in a valid and logical order?
Practicality	One objective is to make sure that the agile levels are practical and can be used in industry. In light of this, to what extent would you agree that these agile levels can be used to classify, and/or aid in the transition of currently employed software development efforts?
Necessity	The classification of agility into 5 agile levels as presented would be beneficial to the software engineering industry?
Relevance	Each of the agile levels presented encompasses a set of agile practices and concepts. To what extent would you agree that those practices and concepts are relevant and correctly assigned to their respective agile levels?
OPEN-ENDED QUESTIONS	
Would you add, remove or redefine any of the 5 agile levels? If so, please explain why.	
Do you have any further comments about the agile levels?	

Table 16. Two-Page Survey Questions about the Sidky Agile Measurement Index

Open-ended questions asked about the possible changes to the framework and prompted the user to provide further comments. In particular, it was important to know whether the five levels sufficiently represented the different stages an organization passes through on its journey to agility, and whether they are in the right order. Their sufficiency was determined by assessing the comprehensiveness of the levels. Since one objective of this research was to define a process to guide and assist coaches and organizations when adopting agility, determining the practicality of the levels was very important as this showed whether the research was fulfilling its objectives. Similarly, the necessity of an agile measurement index, from the agile community's perspective, was an important aspect to gather feedback on. The question on relevance was asked to ascertain each participant's opinion on the relevance of the agile practices assigned to each level.

Questions about the 4-Stage Process

Since the 4-Stage process is the main component of the framework, another page of the two-page survey was devoted to it. In the survey, 15 questions measured by the Likert scale assessed understandability (using 6 questions), practicality, necessity, completeness (using 2 questions) and effectiveness (using 5 questions). The two open-ended questions repeated the questions on the SAMI.

LIKERT – SCALED QUESTIONS		
TO DETERMINE …	**QUESTION(S) USED**	
UNDERSTANDABILITY	**For the topics listed to here designate the degree to which you agree that they are understandable**	**Overall objective of this process framework**
		Discontinuing factors
		5 agile levels
		Project –level assessment
		Organizational assessment
		Gap analysis
PRACTICALITY	**One of our objectives is to make sure that the process framework is practical and can be used in industry. In light of this to what extent would you agree that process framework would be practical and feasible to use?**	
NECESSITY	**The process framework is beneficial to the software engineering industry?**	
COMPLETENESS	**All the necessary components are present in this process framework in order to achieve its overall goal of aiding an organization in the adoption of agile development for its various projects?**	
	The steps and activities in the process framework are organized in a logical and valid sequence in order to achieve its overall goal of an assisting agile adoption?	
EFFECTIVENESS	**To what extent do you agree that each of the components in this process framework is necessary and sufficient for the framework to achieve its purpose?**	**Discontinuing factors**
		5 agile levels
		Project –level assessment
		Organizational assessment
		Gap analysis
OPEN-ENDED QUESTIONS		
Would you add, remove or redefine any components of this process framework? If so, please explain why.		
Do you have any additional comments about the process framework?		

Table 17. Two-page Survey Questions about the 4-Stage Process

Since the primary objective of the 4-Stage process is to provide guidance and assistance to organizations, it was necessary to assess the extent to which participants understood the stages of the process and their functionalities. Feedback about the practicality, necessity, and completeness of the 4-Stage process was essential, just as it was to the SAMI, to establish whether the framework provided the services for which it was designed. To assess the effectiveness of the 4-Stage process, the survey elicited the participants' opinions on the necessity and sufficiency of each stage and the contribution of each to the fulfillment of the framework's overall purpose. Table 17 shows the questions used on the one page, within the 2-page survey, devoted to the 4-Stage process.

5.1.3 Participants

A total of 35 members of the agile community took part in the survey. All of them provided informal feedback at the presentations and 27 filled out the two-page survey on-site after the presentation. The remaining participants asked to mail their survey in at a later time. Only one of the aforementioned surveys was received, making a total of 28 responses, yielding a 78 % overall response rate.

The informal feedback and the two-page surveys were gathered in person from the participants at 10 on-site visits made in various parts of the United States. Some presentation sessions, usually those with the more busy and high profile consultants, were conducted one-on-on. Other presentations were conducted in large agile consulting and development firms, where four or five individuals representing different positions and experiences attended.

Demographic information gathered from the participants included their personal information and their experience with agile development. The two-page survey included a question that asked participants for their position in their company or organization, another question asking them for their years of experience, and then a section with four questions gauging their expertise on agility. These, too, used a

Likert scale of 1 for “not familiar” to 7 for “expert.” Table 18 presents the questions used in the survey to gather this demographic information from each participant.

Official Position / Role :				Years of Experience :			
Please rate your familiarity with Agile software development	**1** **NOT FAMILIAR**	**2**	**3**	**4** **SOMEWHAT FAMILIAR**	**5**	**6**	**7** **EXPERT**
Please rate your highest level of involvement in development efforts that used Agile practices	**1** **NONE**	**2**	**3**	**4** **PARTICIPANT**	**5**	**6**	**7** **LEADER**
How frequently have you aided entities in adopting Agile practices	**1** **NEVER**	**2**	**3**	**4** **OCCASIONALLY**	**5**	**6**	**7** **CONSTANTLY**
Please rate your level of familiarity with general process assessment and/or process improvement	**1** **NONE**	**2**	**3**	**4** **PARTICIPANT**	**5**	**6**	**7** **LEADER**

Table 18. Participant Information on Roles and Agile Experience

An important factor when designing this substantiation effort was to make sure that the participants represented a sample spectrum of the entire agile community with respect to different positions and different levels of experiences. While recognizing that diversity was important, it was also imperative that majority of the participants be agile coaches and consultants, since these would be the primary users of this framework.

When classifying the 28 who responded to the two-page survey based on their roles and positions, the participants fell into three categories:

- *Developers*: Eight of the participants (28%) were junior or senior software engineers, architects, analysts or developers
- *Management*: Eight of the participants (28%) were directors, managers or marketing personnel

- *Agile coaches/consultants*: Fourteen of the participants (44%) were agile consultants, agile coaches, senior XP coaches, chief scientists or CTO's

Based on their years of experience, the participants were also divided into three groups:

- 1-2 Years: Eleven of the participants (39%) had between 1-2 years of experience with agile
- 3-5 Years: Nine of the participants (33%) had between 3-5 years of experience with agile
- 6-12 Years: Eight of the participants (28%) had between 6-12 years of experience with agile

Table 19 shows the number of participants by role and experience and highlights the relation between the two categories.

Groups	1-2 Years	3-5 Years	6-12 Years	Total
Agile Coaches	2	5	5	12
Management	3	2	3	8
Developers	6	2	0	8
Total	11	9	8	28

Table 19. Categorization of Participants by their Roles and Years of Experience

A couple of interesting observations concerning the grouping of the participants are as follows:

- Most of the agile coaches have 3 or more years of experience
- Most of the developers have 1-2 years of experience
- Management is distributed equally in terms of years of experience

Also, of the 28 participants, five people made up a special group that was highly experienced with process assessment and process improvement efforts, and have led agile adoption efforts for over six years. Since this group represents the experts and leaders of the agile adoption community, their participation was important to the success of this validation effort. They possess substantial knowledge about agile adoption and are, therefore, better able to assess the overall performance of the Agile Adoption Framework. The remaining participants are also important to the validation effort, because they assess how the framework meets their need for guidance throughout the adoption process. As a total group, the participants represent the range of experience usually encountered as organizations embark on the journey to agility.

The previous sections have provided an outline of the three main components of the study, the procedure, the questions, and the participants. The next section presents an analysis of the quantitative feedback gathered from the agile community.

5.2. Quantitative Feedback

Due to the nature of the feedback sessions and the availability of members from the agile community, the number of participants is relatively small for complex statistical analysis. Therefore, the results presented in this section are based on simple statistical descriptions, data trends and comparisons of percentages. The results related to each of the components of the framework are presented separately, starting with the SAMI.

5.2.1 Results concerning the Sidky Agile Measurement Index (SAMI)

Figure 33 depicts the overall results of the one page of the two-page survey assessing the SAMI. The figure shows that over 75%, representing a majority of all the respondents, either believe or strongly believe that the SAMI is comprehensive, practical, and necessary. However, the response to the question on the relevance of the agile practices within the agile levels shows the agreement rate drops to a little below 50%, while the rate of disagreement rises to approximately 37%; the rest neither agree nor disagree. Figure 34 shows a breakdown of the overall data by the experience of the participants for further analysis of the data for each aspect of the levels being assessed. Figure 35 shows a breakdown of the data by role (developers, management and coaches/consultants).

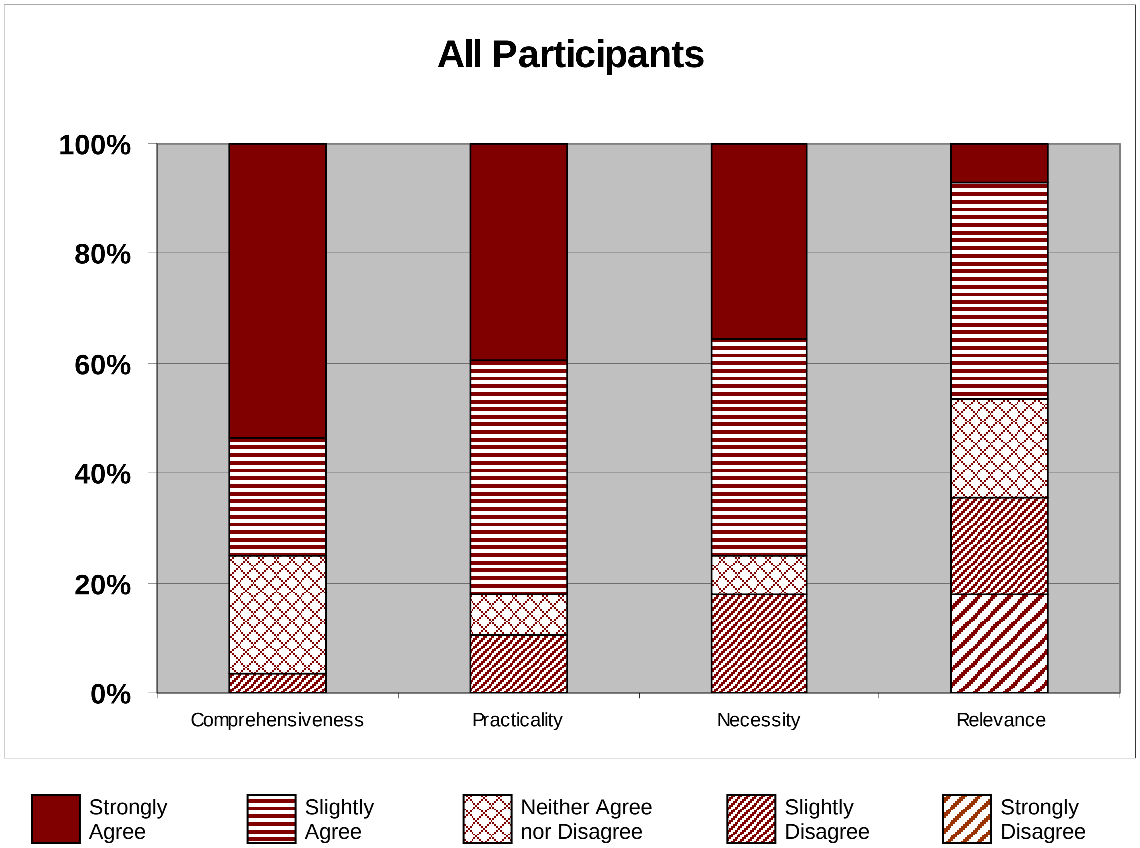


Figure 33. Overall results for the SAMI

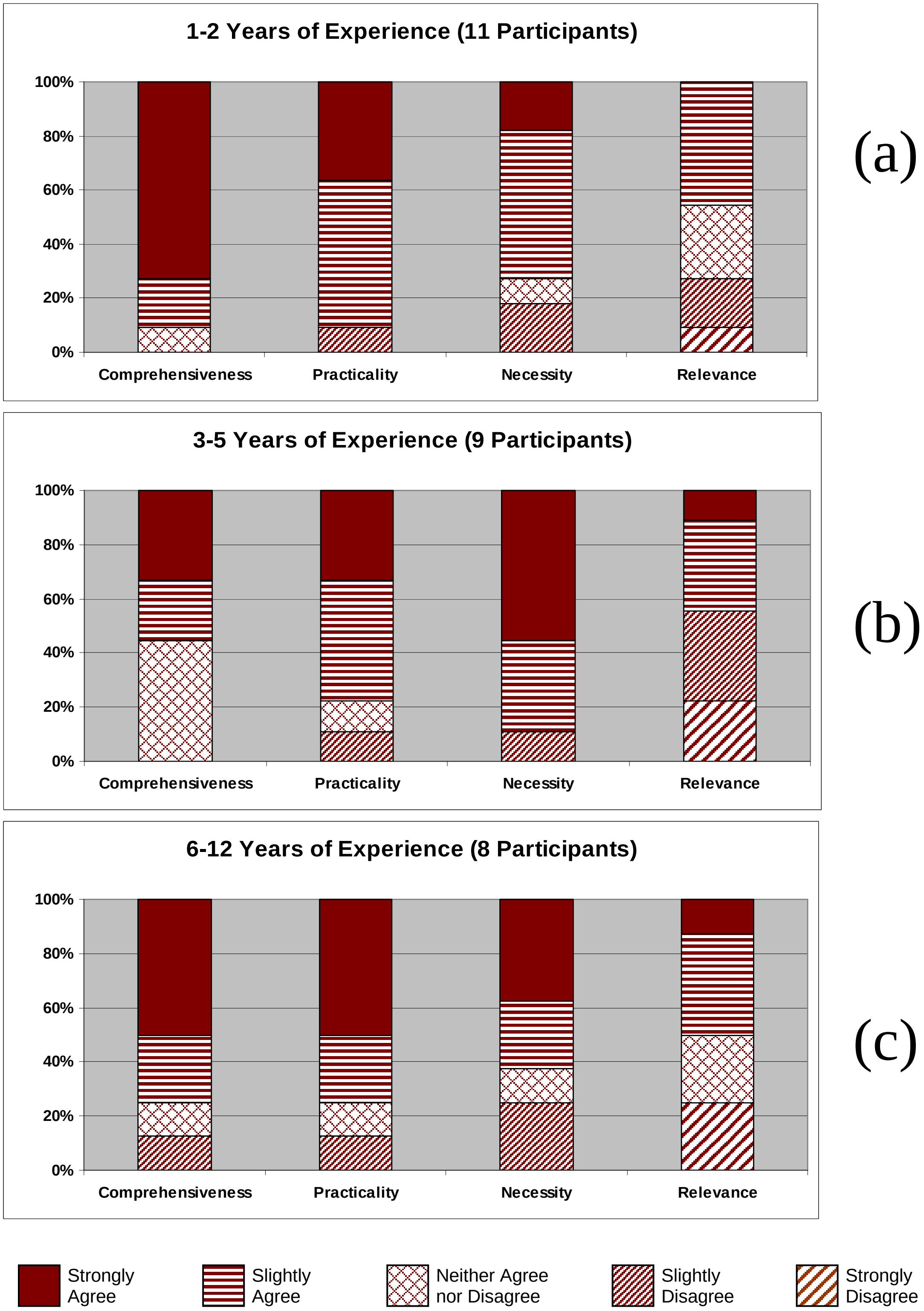


Figure 34. Results of the SAMI Categorized by Years of Experience

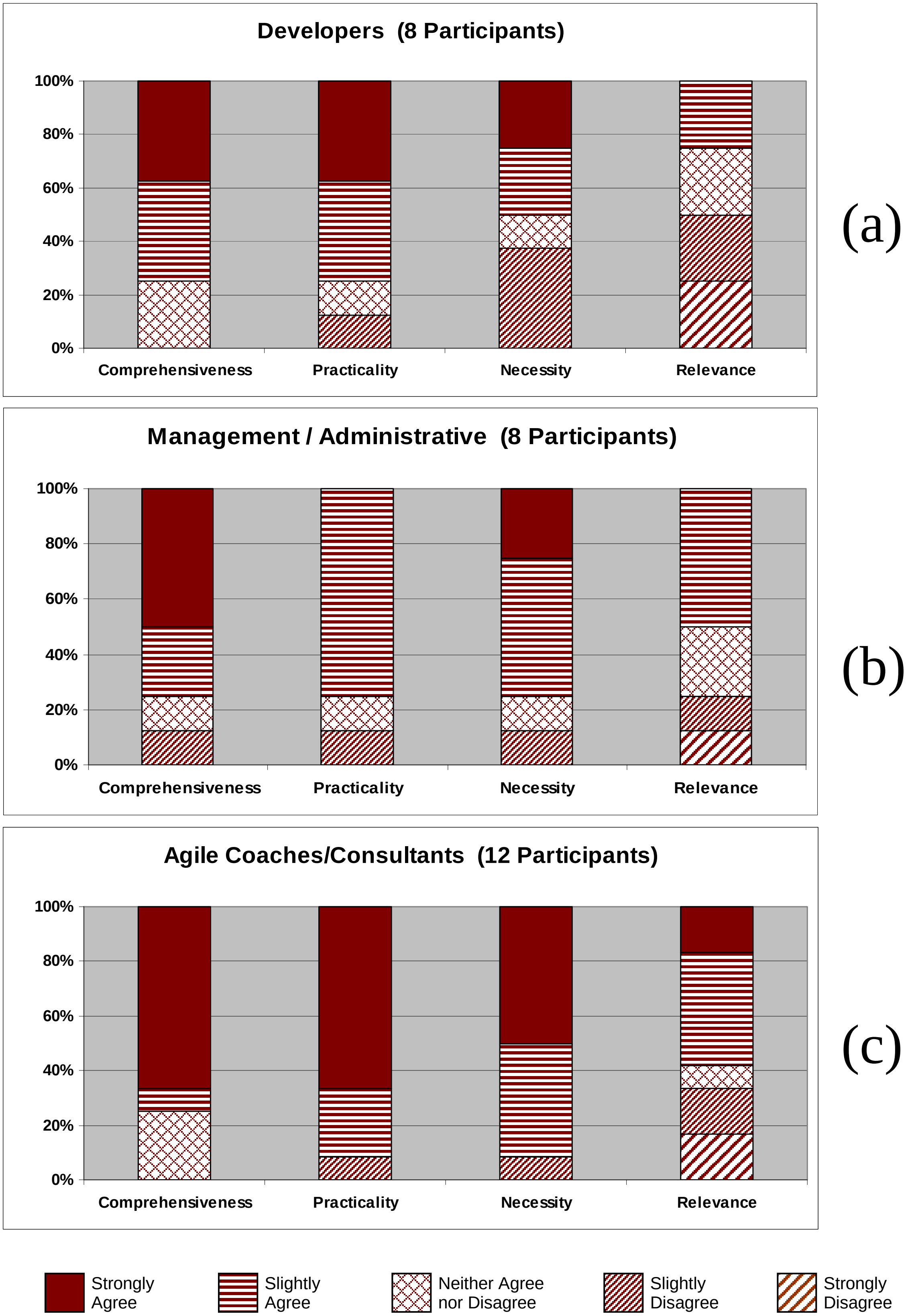


Figure 35. Results of the SAMI Categorized by Roles

Comprehensiveness

As Figure 33 shows, 53.5% of the participants strongly agreed, 21.5% slightly agreed, 21.5% neither agreed nor disagreed, and only 3.5% slightly disagreed that the 5 levels are comprehensive enough to reflect the various steps most organizations go through to reach agility. The overall mean for all the participants was 4.05 (slightly agree), with a standard deviation of 0.89.

As Figure 34 shows, when categorized by years of experience, the results exhibit a trend: both sides of the experience spectrum were more in agreement than the middle. Participants with 1-2 years (Figure 34a) and with 6-12 years (Figure 34c) of experience have a close by agreement rates (90% and 75% respectively) about the comprehensiveness of the levels, probably because the group with less experience (90%, Figure 34a) did not have enough experience to compare the levels with real life experiences. At the opposite end of the spectrum, participants with extensive experience (75%, Figure 34c) have been involved with many projects and have agreed that the levels are comprehensive enough to capture the steps organizations go through to reach agility. Representing the middle ground are participants with 3-5 years of experience (Figure 34b) who were 45% neutral and 55% in agreement with the comprehensiveness of the levels. The closeness of agreement between participants with 1-2 years and 6-12 years of experience is significant because the former group represents the need for guidance to agility and the latter group has the experience to judge adequacy of the completeness of the Agile Adoption Framework

As Figure 35 shows, when categorized by role, the results exhibit a different trend in that 75% of developers, management and coaches agreed on the adequacy of the comprehensiveness of the five levels of agility. Usually developers (Figure 35a) are not involved with the overall picture of adopting agility in organizations and the levels it goes through, and hence might have agreed because of inexperience (they have nothing to compare it to). However, having 75% of the agile coaches/consultants (Figure 35c) and managers (Figure 35b) agree with the levels' comprehensiveness is significant because the nature of their work makes them familiar with the agile adoption process.

In summary, the survey participants perceived the SAMI as sufficient and comprehensive enough to represent the different stages an organization goes through to achieve agility. The positive feedback on the comprehensiveness of the levels of agility is crucial in the substantiation of the overall framework, because an essential part of the 4-Stage process (Stages 2 and 3) relies on the structure and priority of the levels of agility in determining the target level of agility that projects can adopt, and the readiness of the organization for adoption. Therefore, when the agile community agrees that the levels represent the right steps and order of an organization's move towards agility, the validity of the whole framework increases.

Practicality

Out of all the participants, Figure 33 shows 39.3% strongly agreed, while 42.9% slightly agreed that the SAMI is practical and can be used during agile adoption efforts. Of the rest, 7.1% neither agreed nor disagreed and 10.7% slightly disagreed. The overall mean is 4.1 (slightly agree), with a standard deviation of 0.95.

As all the parts of Figure 34 show, years of experience did not impact the practicality results. At 90% for 1-2 years of experience, 77% for 3-5 years, and 91% for 6-12 years, the percentages of agreement were almost the same regardless of the level of experience. However, when classified by roles, the results did vary. While 39% of the developers (Figure 35a) strongly agreed on the practicality of the Agile Adoption Framework, almost 30% were divided between neutral or slightly disagreeing. This can be attributed to the fact that developers usually do not use the levels of agility during adoption efforts. Not a single manager strongly agreed with the practicality of the levels, yet almost 80% of them (Figure 35b) slightly agreed to it. This reflects the way managers are involved in the process. They are usually the ones who have to sell these techniques to the clients, and it is hard to sell something without having full confidence in it and having seen it working. However, the managers see the potential of the framework, as the 80% in slight agreement indicates (Figure 35b), but are still skeptical. The agile coaches are more than 90% (Figure 35c) in agreement that the levels are practical and can be used during agile adoption. Unlike developers, agile coaches and consultants usually use the levels of agility during adoption efforts. Therefore it was not surprising that they exhibited more confidence in the practicality of the framework due to their hands-on experience and their understanding of the usefulness of the agile levels to the adoption process.

Necessity

As Figure 33 shows, 35.7% of all the participants strongly agreed that the SAMI is necessary, and 39.3% slightly agreed, while 7.1% were neutral and 17.9% slightly disagreed. The overall mean is 3.9 (slightly agree), with a standard deviation of 1.08.

The agreement levels for necessity are slightly lower than those for comprehensiveness and practicality because of the developers' responses to this question. Although approximately 50% agreed to the necessity of the levels of agility, the high level of disagreement occurred, because developers have no need or use for an agile measurement index in their day-to-day work. They are not involved with actual agile adoption efforts. However, high level of agreement (over 75%) from management (Figure 35b) and coaches (Figure 35c) reflects the need for an agile measurement index within the agile community.

Relevance

Figure 33 illustrates that only 7.1% of the participants strongly agreed that the agile practices were assigned to relevant agile levels. However, 39.2% slightly agreed, while 17.9% neither agreed or disagreed and another 17.9% slightly disagreed.

However, for the first time, 17.9% of the participants strongly disagreed. The mean for the relevance was 3.0 (neither agree nor disagree), with a standard deviation of 1.27.

The low agreement rate for relevance was expected and accounted for in the framework. The overall results of the comprehensiveness, practicality, and necessity categories seen in the one page of the two-page survey devoted to the SAMI highlights the agreement of the agile community on their utility. However, the agreement rate is much lower when it comes to the details of the levels of agility and where to place each agile practice or concept, because some of the participants lacked actual experience in the journey to agility and those with experience, especially consultants and coaches, have developed their own methods for making the transition. The problem is that these methods more than often are based on the means and structure of a specific organization, and therefore, are not always applicable to other organizations.

The five levels of agility solve this problem because of its main contribution to the agile adoption process – the repeatable structure and guidance that can be used for agile adoption effort. Agile coaches can tailor the levels of agility and place agile practices in different levels to reflect their experience and approaches to agile adoption. The low agreement rate on the details of the levels does not jeopardize the validity of the framework. It simply testifies to the fact that each person in the agile community has his or her own opinions concerning when and why to introduce different agile practices. Nevertheless, the results from the two-page survey on the other aspects show that there is in need for structure and guidance on how to organize these agile practices and concepts; and this is exactly what the SAMI provides.

The Experts' Response to the Sidky Agile Measurement Index

The overall results, and the clarifications by roles and years of experience raise the concern of what the experts had to say about the agile adoption framework. This section, therefore, presents only the results obtained from the special group of agile experts defined earlier in this chapter.

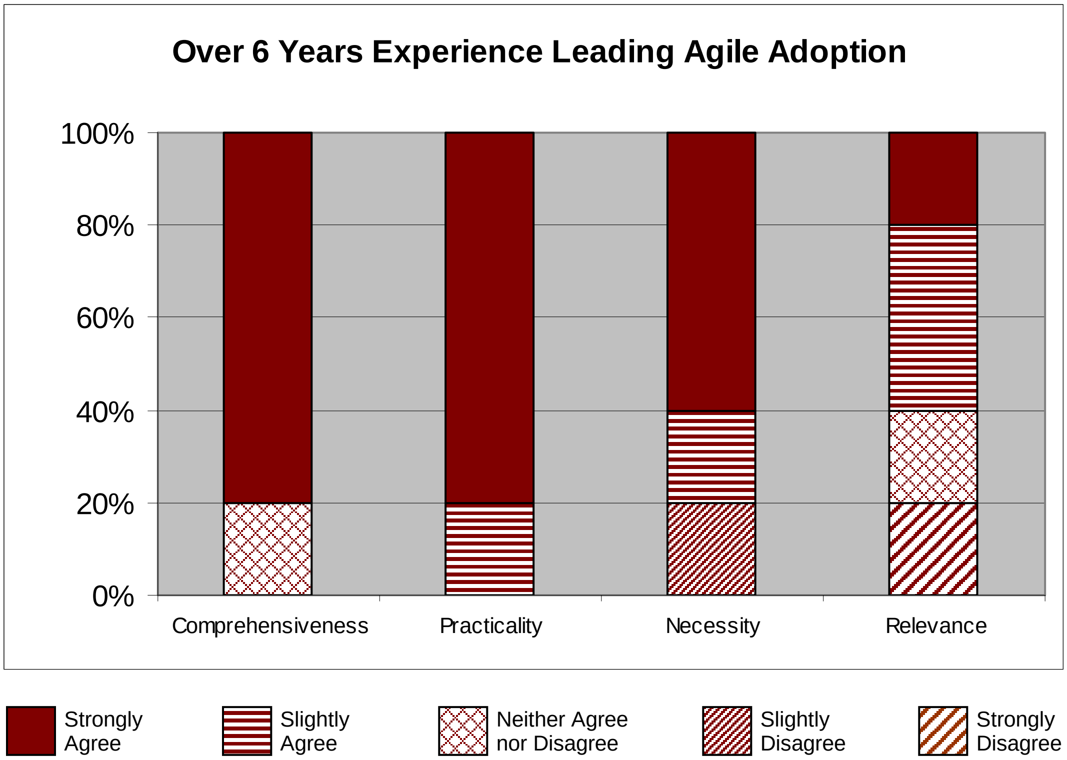


Figure 36. Results of the Agile Measurement Index from agile experts

As Figure 36 shows, 80% of these experts strongly agreed with the comprehensiveness of the agile levels, while the remaining 20% neither agreed nor disagreed. They all agreed to the practicality of the levels of agility, with 80% indicating strong support. Also, 80% agreed to the necessity of the levels of agility and 20% slightly disagreed. As for the relevance, 60% agreed that the practices are more or less in the right levels and 20% chose to remain neutral until they have studied the 5 levels more thoroughly. The remaining 20% strongly disagreed.

Overall, the agile community has accepted the SAMI. This is important because they are the foundation used by the 4-Stage process. The next section presents the feedback obtained from the second page, of the two-page survey, which is devoted to gathering feedback about the 4-Stage process.

5.2.2 Results concerning the 4-Stage Process

As Figure 37 shows, the majority of all the participants (approximately 80%) either agreed or strongly agreed with all five aspects of the 4-Stage process. What is encouraging is that not a single participant strongly disagreed with any aspect of the 4-Stage process, and only one participant slightly disagreed with its completeness.

Figure 38 provides a breakdown of the participants by years of experience and Figure 39 by role. The next sections present an analysis of these groupings.

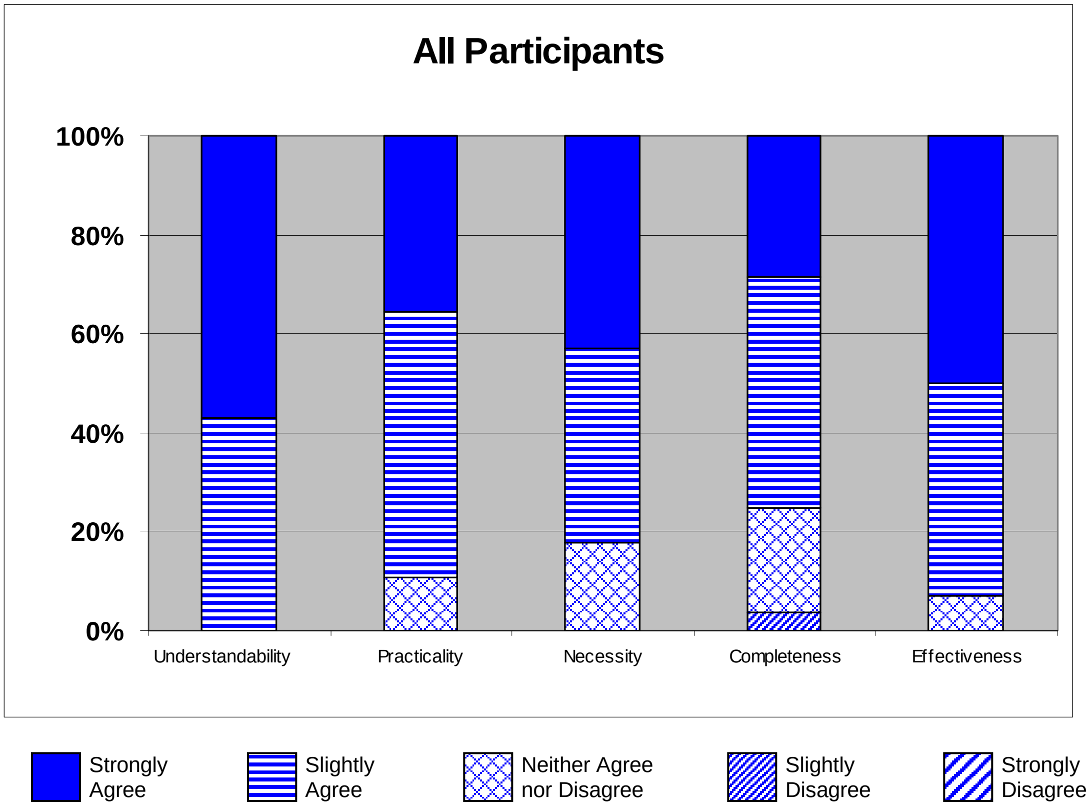


Figure 37. Overall results for the 4-Stage Process

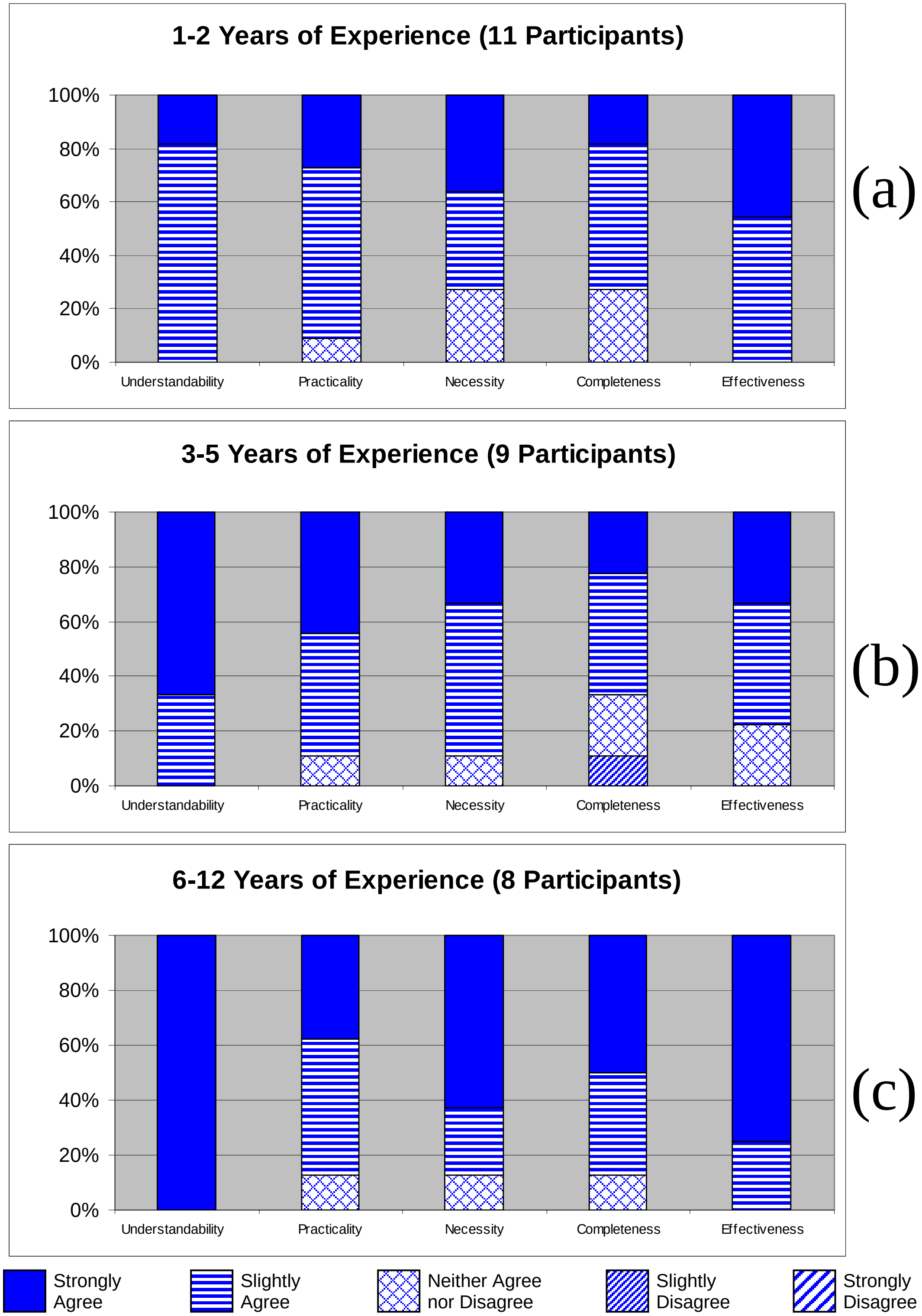


Figure 38. Results of the 4-Stage process categorized by experience

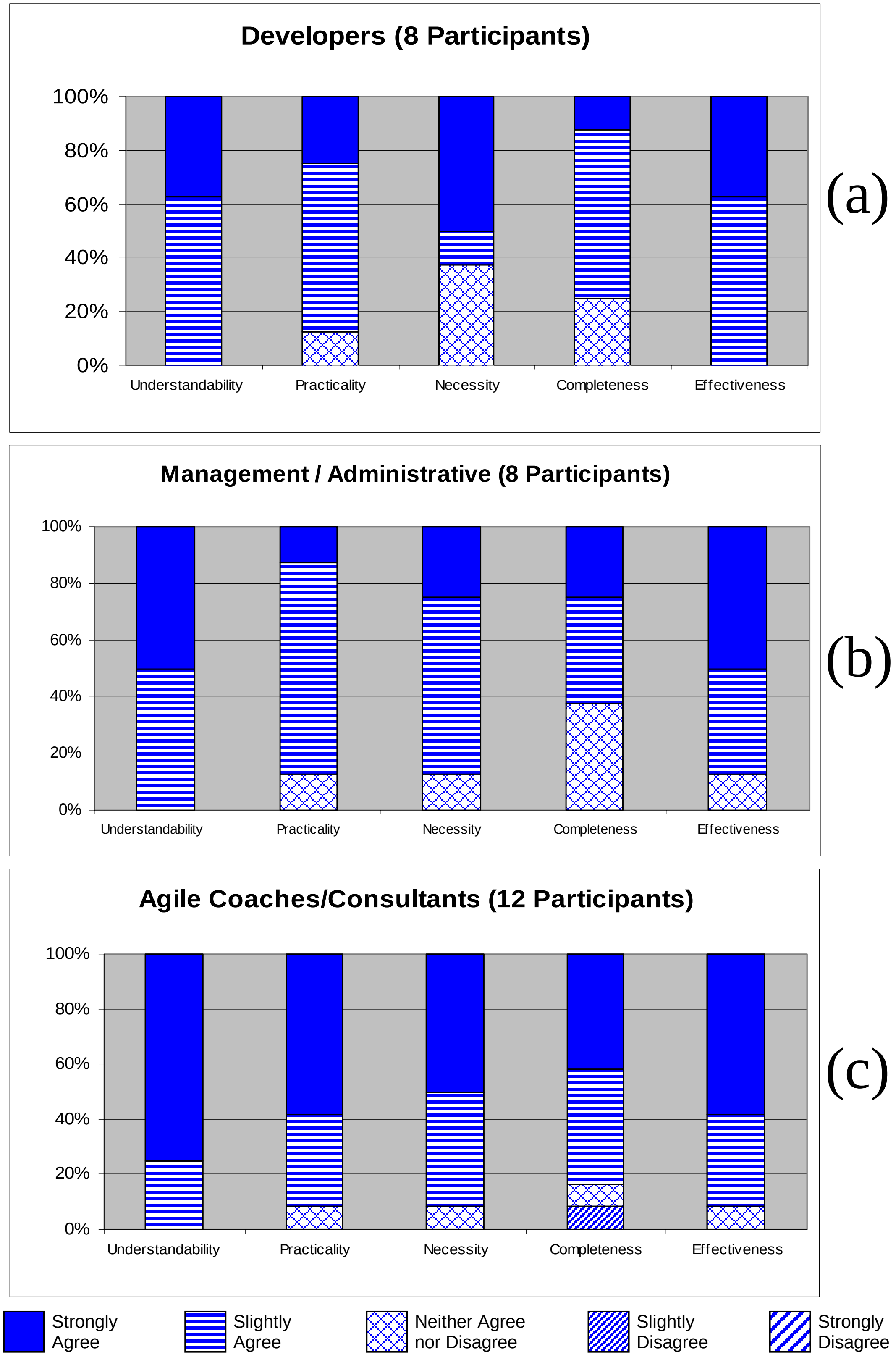


Figure 39. Results of the 4-Stage process categorized by role

Understandability

As Figure 37 shows, everyone understood all the stages of the process, with 57.2% of the participants strongly agreeing to the understandability of the four stages of the process, and 42.8% slightly agreeing to it. The overall mean for all the participants is 4.48 (between slightly and strongly agreeing), with a very low standard deviation of 0.49.

When categorized by years of experience, as Figure 38 shows, a trend emerged: the more experienced the participant was, the more strongly he or she agreed that the process is understandable. As Figure 39 shows, the results categorized by role also reflect this trend. Since the 4-Stage process provides guidance about agile adoption efforts, it is logical, and expected, that the agile coaches would understand the process more than the developers and managers.

Practicality

While none of the participants disagreed about the practicality of the 4-Stage process, as Figure 37 shows, the strong consensus seen for understandability diminished, with 35.7% of the participants strongly agreeing, 53.6% slightly agreeing and 10.7% remaining neutral. The overall mean is 4.25 (a little above slightly agree), with a standard deviation of 0.64.

Interestingly, the results were almost the same regardless of the years of experience, as all parts of Figure 38 show. However, when classified by roles, the results did vary. The developers and managers were slightly more hesitant to agree to the practicality of the 4-Stage process. This observation is seen in Figure 39a and Figure 39b, only 25% the developers and 12.5% of the managers strongly agreed while 62.5% of the developers and 75% of the managers slightly agreed to the stages' practicality. The agile coaches and consultants, on the other hand, were much stronger in their agreement to its practicality. Figure 39c shows that 58% of the coaches strongly agreed while only 34% of the slightly agreed. One possible reason behind this is that since the agile coaches are the ones involved with the adoption process, they have a greater ability than the managers and developers to grasp the practicality of actually using the 4-Stage process to guide the adoption efforts.

Necessity

Returning to Figure 37, 42.9% of all participants strongly agreed, while 39.3% slightly agreed that agile community needs the 4-Stage process and 17.8% neither agreed nor disagreed. The overall mean is 4.25 (a little above slightly agree), the same as the practicality mean, but the standard deviation of 0.75 is a little higher.

The agreement levels for necessity are slightly lower than for practicality. Like the drop in agreement seen for the agile levels, this drop reflects the developers' responses. The developers (see Figure 39a) were hesitant to make up their minds about the necessity of the 4-Stage process (37.5% neither agree nor disagree). Their hesitation can be attributed to their non-involvement with agile adoption efforts and, therefore, not deeming a structured approach necessary. The high level of agreement from management seen in Figure 39b (87.5%) and from coaches (91.6%) seen in Figure 39c, the two groups most often involved in the actual adoption efforts, reflects the need for a structured approach to agile adoption.

Completeness

As Figure 37 shows, the percentage of participants who strongly agreed all the necessary components to guide an adoption effort are within the 4-Stage process was only 28.6%, while 46.4% slightly agreed, 21.4% neither agreed nor disagreed, and 3.6 percent slightly disagreed. The mean for the completeness was 3.8 (a little below slightly agree), with a standard deviation of 0.78.

The completeness aspect of the 4-Stage process had the lowest agreement percentage of all the aspects of the process. Contributing to this was the actual process used to gather the feedback. The 90 minutes allotted for presenting the whole framework to the participants, subsequent discussions, and the survey was simply not enough time for someone to grasp the essence of the whole framework and attest to its completeness without actually sitting down and going over all its components carefully. Moreover, the presentation shown to the participants was on a macro level view due again to time constraints. The participants that returned the surveys at a later time validate these observations. This is because with time to look over the framework in detail, both strongly agreed that the 4-Stage process is complete.

Effectiveness

In Figure 37, it also shows that the agile community gave a positive response to each of the components of 4-Stage with regards to its effectiveness (i.e. achieving what it was designed for). Fifty percent (50%) of them strongly agreed, and 42.9% slightly agreed, while only 7.1% were neutral. The overall mean is 4.43 (between slightly and strongly agreeing), with a standard deviation of 0.63.

Although all parts of Figure 38 and Figure 39 show no strongly defined pattern emerging for the effectiveness of the 4-Stage process, in general the results support a positive perception of its effectiveness, with most of the participants, no matter their role or experience level, expressed agreement. Only a few were neutral, neither agree nor disagree, while 75% of the most experienced participants (see Figure 38c) showed strong agreement

The Experts' Response to the 4-Stage Process

To conclude the analysis of the quantitative results, this section presents the feedback on the 4-Stage process gathered from the agile experts.

As Figure 40 shows, what is truly noteworthy about the results gathered from surveying the experts is that they agreed 100% to all the aspects of the process. These results underscore the perceived utility of the agile adoption framework and substantiate its validity.

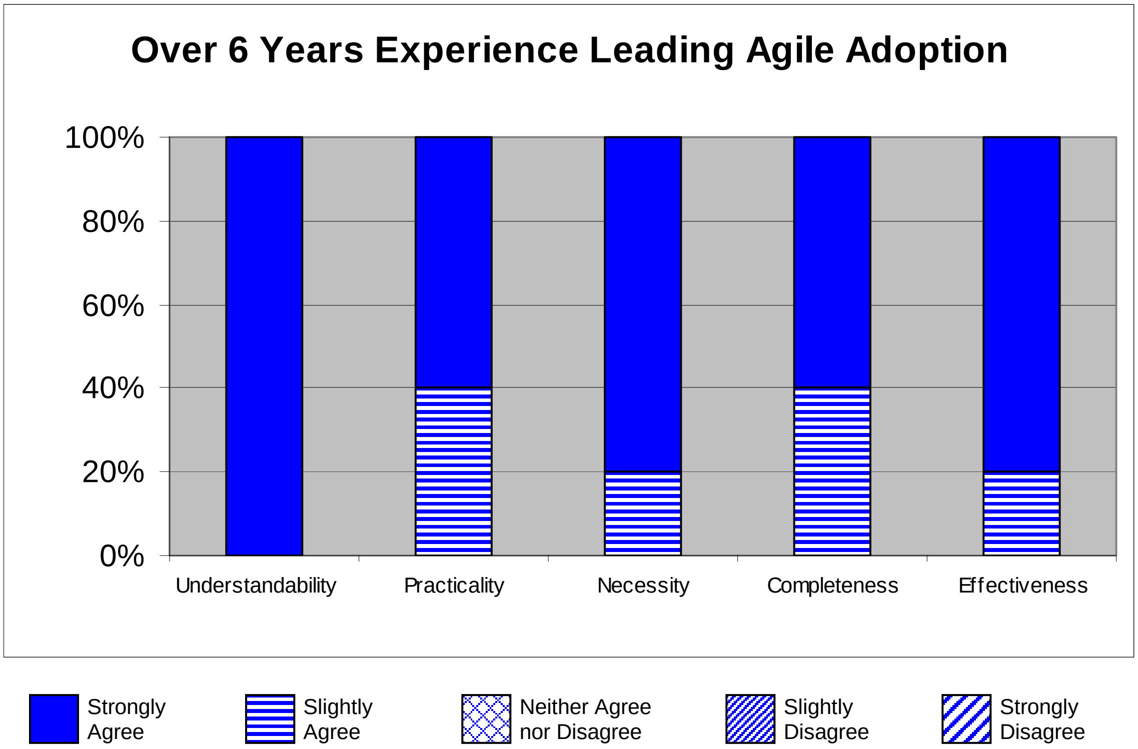


Figure 40. Results for the 4-Stage process from Agile Experts

The next section provides a discussion of the qualitative feedback gathered from the agile community.

5.3. Qualitative feedback

Despite the fact that only 78% of the participants (28 people) contributed to the quantitative feedback (via the 2-page survey), all 35 participants provided qualitative feedback. The three sources for the qualitative feedback are the:

- open-ended survey questions from the 2-page survey
- informal discussions that occurred during the feedback sessions
- 12-page surveys that were returned

Since during the presentation the participants expressed their many of their concerns through the informal discussions, many of them left the open-ended questions empty as the issues had already been discussed earlier. Consequently, reporting the results of the open-ended questions separately is not possible. We combined all the qualitative feedback together, whether they were written or discussed, and grouped them could together based on the feedback topic.

The results of the 12-page survey are included in the qualitative feedback section because only two 12-page surveys were returned making it impractical to conduct any statistical analysis for the quantitative aspect of the results. Therefore, all the feedback obtained through these surveys is included in this section.

Due to the general consensus of the participants on the validity of the 4-Stage process, and the qualitative feedback on this component of the framework tended not to raise problems that needed to be addressed. Therefore, most of what follows focuses on the SAMI and is divided based on the different topics of the qualitative feedback.

5.3.1 The Sidky Agile Measurement Index

Most of the discussions revolved around the agile measurement index. There were many beneficial insights and spirited discussions about the notion of agile levels. Each of the following subsections presents the feedback related to one aspect of the agile measurement index.

The Agile Levels and the CMM

While many participants favored the notion of agile levels, and an agile measurement index, other participants expressed concern as soon as they read the words “agile levels” and found out that there are five. Immediately the CMM came to their minds. Since many participants apparently did not like the CMM, there was an immediate negative reaction towards the levels of agility. Some of the participants believed that the CMM causes more harm than benefit to development processes, and that it has become a mere certification process, which is hard to maintain in reality. However, when the focus of the discussion shifted from a comparison of the agile levels and the CMM to the functionality of the agile levels, and the fact that they are assigned to individual projects and not the whole organization, the participants viewed the agile levels more favorably.

The Business Value

During these informal feedback discussions, several participants mentioned that the agile levels should be linked, or related to, the business value realized when the level is attained. This point has been mentioned and addressed earlier in the discussion of the tailorability of the SAMI. (See Chapter 4)

Another beneficial suggestion related to the business value and agile levels was to have the agile levels populated with different agile practices based on the business value the organization wishes to attain. For example, if the business objective of an organization is to have a shorter time-to-market, then the SAMI would have the same levels and principles, but all the practices and concepts would be related to achieving a shorter time-to-market. This is basically adding a third dimension, the business value or objective, to the SAMI.

The Terminology of the Agile Levels

Due to the CMM, the phrase “agile levels” had a negative connotation for some of the participants. They agreed with the functionality in the purpose of the agile levels, but the name was annoying to them. They suggested replacing “agile levels” with, “agile dimensions,” “agile sets,” or “agile regions”.

Other participants accepted the term “levels” but wanted to change the names of the actual levels. They wanted the level names to reflect the type of the practices within the level instead of the value achieved when adopting the practices within a level. Therefore, they suggested that the *Evolutionary Level* be named the *Timing Level,* because all the practices within the level focused, in one way or another, on the frequency of iterations and releases. They also suggested that the *Effective Level* be named the *Engineering Level,* and that the *Encompassing Level* be named the *Environment Level.* In contrast, others explicitly praised the naming of the levels.

This feedback concerning the names of the agile levels was carefully analyzed and discussed. The result was that one of the names of the agile levels was changed. Level 5, which was originally named “*Optimal*,” was changed to “*Encompassing,*” because encompassing seemed to better reflect the essence of the level, which is to foster a vibrant environment to sustain, foster, and spread agility in the whole organization.

Also, one of the co-authors of the agile manifesto recommended the change of a single word. Originally, the second agile principle “plan and deliver software frequently,” was written as “plan to deliver software frequently.” There is a subtle, but fundamental difference between the two statements. Planning *to* deliver software does not necessarily ensure the delivery of the software. Therefore, the more appropriate wording for the principle was “plan *and* deliver software frequently.”

Organization of Practices

Of course the organization of the agile practices within the agile levels provoked considerable discussion. Some participants just had gut feelings, based on their experiences, that the practices needed to be shuffled around. Others argued that practices found in the higher levels needed to be in the lower levels, because they created the conditions that made other practices easier to adopt. These are issues of tailorability, and the framework provides the structure and guidance to populate the levels of agility, but does not dictate the organization of the practices within the levels.

As a result of all the conversations and discussions about reorganizing the agile practices, the location of one practice, *reflect and tune the process,* was moved from level 4 to level 1. The decision to make this change is an acknowledgment that reflecting and tuning the process form one of the cornerstone concepts in agility and contribute to enhancing communication and collaboration within the project team.

An important concern: the tailorability of the SAMI

One of the leaders in agility had a deeper concern with the SAMI as presented. Although he did not fundamentally disagree with the positioning of the practices within the levels (of course he had his preferences) and understood the flexibility given by the framework to reorganize the practices, nevertheless he was concerned that the 5 Levels of Agility (along with their practices and indicators) are currently the only instance of the SAMI. He was afraid that beginners within the agile community might look at the five levels, along with the current organization of agile practices within them, and think that that is the only way and only arrangement for agile practices within agile levels. This is a valid and important concern that all the publications have addressed by stressing the fact that the SAMI presented in this research is only one instance of the SAMI, which is tailorable, and that other instances can exist.

5.3.2 The Discontinuing Factors (Stage 1)

Another topic in several of the feedback sessions was that of the discontinuing factors. The two issues discussed were the appropriateness of two factors and the naming of the factors. The following two subsections elaborate on these issues.

Factor Appropriateness

The first discontinuing factor identified in the agile adoption framework is the *Inappropriate Need for Agility.* Originally, this factor referred to the situation where an organization delivered products on time and within budget, the rate of change of requirements in the project was low, and there was no need for short time-to-market values. In other words, adopting the agile practices would add no value to the software development process, because there were no problems. The survey participant, who was a co-author of the Agile Manifesto, remarked that many people, including himself initially, believed that in such situations there was no need for adopting agility. However, he then said that he recently discovered many organizations, including ones he had worked with, were in such situations and still adopted agility. He concluded that people adopt agility not only when there is a problem with their current process, but just to optimize their development process and increase efficiency. Essentially this relates to the notion of incorporating business value, because internal optimization is a business value many organizations seek. Therefore, with this new piece of information, the first discontinuing factor was redefined to include the new realizations.

Originally the framework had four discontinuing factors. The fourth discontinuing factor was if the project developed was either mission or life-critical. The participants agreed that a couple of years ago there was still a lot of speculation about whether agility was suitable for mission and life-critical systems. However, in light of the current status of agile software development, the majority of the participants disagreed that developing mission and life-critical systems would be discontinuing factor. These opinions were based on the fact that many of them had personally been involved with projects developing such systems that used agile practices.

The Titles of the Discontinuing Factors

The discussions and surveys also highlighted the need for more expressive titles of discontinuing factors. Originally, the first discontinuing factor was titled "*No Value Added.*" Although, most participants did not disagree with this title, it did confuse them. After further explanation of this discontinuing factor, participants suggested better titles that would be more expressive. After due consideration, the title of the first discontinuing factor was changed to "*Inappropriate Need for Agility.*"

A small number of the participants, all from the same organization, suggested a new set of titles for all the discontinuing factors, while adding some of their own. The discontinuing factors they suggested were:

- No need
- No support
- No money
- No permission
- No Courage

5.3.3 Other Comments

Several other miscellaneous comments pertaining to different areas of the framework are grouped together in this section. Each of the following subsections presents one of these comments

The Validity of an Agile Practice

The feedback on one agile practice, *"the use of true object oriented design and construction"* highlighted a disagreement among agile practitioners. While some participants believed that most of the development done in agility is based on object oriented design and construction, other participants felt very strongly that this is not a practice that is related to agility. They argued convincingly that agile software development can be used with other design and construction paradigms. As a result of these discussions, and further research, the decision was made to remove this agile practice.

Why a Target level of Agility

A small number of the participants expressed concern that determining a target level for a project would limit and restrict that project's agile potential. For example, even if the project is incapable of adopting all the practices in level 3, why discourage it from the rest of the practices of levels 4 and 5. The answer to this concern goes back to the design of measurement index. Each agile level is composed of a certain set of agile practices that work in synergy to introduce a particular agile value into the development process. The practices are organized and laid out based on the agile principles. Within each principle, the practices and concepts build on and complement each other. If we consider the agile principles as the pillars that hold up the agility in the organization, then whole building (development process) is only as strong as its weakest pillar. Similarly, if all the agile practices from level 1 to 5 for all the agile principles except one are adopted, then overall the project is not fully agile. It is only as agile as its weakest principle.

Therefore, when factors outside the control of the organization constrain the highest level of agility for a project, the focus should be on resolving the constraining factors so that the all the principles of agility can rise to higher levels of agility. This broader approach is better and more beneficial than focusing on ways to adopt agile practices in higher levels, because constraining factors will keep one of the principles at a lower level of agility than the rest, thereby weakening the overall agility of the project.

Overall Terminology

The three roles frequently mentioned in the framework are developers, managers, and customers. Many of the participants wanted a better and clearer definition of these roles and several even suggested different definitions for these roles. One participant defined the roles using an interesting metaphor worth sharing: developers are the *geeks* who make decisions of technical value, customers are the *suits* who make decisions of business value, and managers are *doctors* who take care of the health of the entire community.

This chapter has presented and analyzed the feedback gathered from the industrial agile community used to substantiate the validity of the agile adoption framework. Overall, based on the quantitative results obtained from 28 surveys, the feedback was positive and shows the communities in agreement as to the utility of this framework. The qualitative feedback has focused on highlighting the constructive criticism received from industry and explaining how it has been accommodated.

6. Conclusion

The creation of the Agile Adoption Framework, presented in this dissertation, is motivated by the absence of a structured approach capable of providing organizations with guidance on how to adopt agile practices. Organizations are seeking methods to help determine whether they are ready for the change to agility and to identify the right agile practices to adopt. In addition, organizations are interested in learning more about the preparations necessary for and the potential difficulties in their journey toward agility.

The Agile Adoption Framework is a first step toward addressing the need for providing organizations with a structured and repeatable approach to guide and assist them in their journey toward agility. The Agile Adoption Framework provides organizations with guidance through a structured and well-defined 4-Stage Process. While the 4-Stage Process is the key component of the framework, it relies heavily on another important component in the framework, the Sidky Agile Measurement Index (SAMI). The SAMI is the measurement index that supports the measuring of the agile potential of projects and organizations, using the notion of Agile Levels. Each Agile Level consists of a set of Agile Practices categorized according to the Agile Principle they manifest. Moreover, a set of Indicators accompany each Agile Practice. These indicators are used to assess the organization's readiness to adopt the Agile Practice they are associated with.

The first stage (Identifying Discounting Factors) within the 4-Stage Process concentrates on conducting a pre-adoption assessment to determine whether the organization is ready for the move to agility or not. The pre-adoption assessment focuses on determining the degree of presence of discontinuing factors. When discontinuing factors are found, like the absence of executive support, or the lack of sufficient funds, then the agile adoption process is suspended until the discontinuing factors are removed.

By utilizing the SAMI, the second stage (Project Level Assessment) determines the agile potential for a specific project. The agile potential of a project is the highest level of agility the project can reach. Project agility is determined by assessing factors needed for the adoption of agile practices which the organization has no control over changing. On the other hand, stage 3 (Organizational Readiness Assessment) assesses the degree to which the organization is ready to support the project in reaching its agile potential. This is determined by assessing the organization's readiness to adopt each agile practice in the project's target agile level, as defined by the SAMI. The fourth stage (Reconciliation) of the 4-Stage Process reconciles any differences between the project's agile target level and the organization's readiness level. The result is a recommended set of agile practices for the organization to adopt.

Both components of the Agile Adoption Framework, the 4-Stage Process and the SAMI, were presented to over 35 members of the agile community to gather feedback about their "goodness." The feedback obtained from this substantiation process is promising and conveys the perceived effectiveness of the framework along with each of its components.

While the Agile Adoption Framework provides organizations with guidance and assistance regarding a number of issues related to agile adoption, it does not guide the complete process improvement lifecycle of the organization. The Agile Adoption Framework focuses on providing guidance in the initial stages of process improvement lifecycles like IDEAL. The Agile Adoption Framework has a strong assessment focus in order to prepare organizations before attempting to adopt agile practices. At the same time, as explained in the 4-Stage Process, the assessment conducted by the framework also provides the organization with guidance regarding what changes need to occur in the organization to prepare for the adoption of certain practices. With the help of the SAMI, the framework also provides the organization with a roadmap that illustrates the steps (i.e. levels) necessary to start the agile adoption effort. The framework provides this guidance independent of any one particular agile method or style. That is, the framework is not based on XP or SCRUM or any other agile style; it is based on the core values and principles of agility as defined by the Agile Manifesto.

6.1. Main Contributions

The Agile Adoption Framework has contributed to the Computer Science body of knowledge, and in particular to that of Software Engineering and process improvement related to agility. The framework's contributions are manifested through providing the agile community with a structured approach to agile adoption and a measurement index for agility. These contributions originate from the two components of the framework, the 4-Stage Process and the Sidky Agile Measurement Index.

The 4-Stage Process: provides a structured approach that guides and assists organizations seeking to adopt agile practices. Moreover, the four stages are sequenced in an effective manner that avoids conducting unnecessary activities, while at the same time ensuring that neither the project's potential nor the organizational characteristics are overlooked. The 4-Stage process provides a number of benefits and contributions, mainly:

- *Structuring the Agile Adoption Process*

  The 4-Stage process introduces structure in the complex and unpredictable process of the agile adoption. The structuring of the process is divided into four stages: (1) Identifying Discontinuing Factors, (2) Project Level Assessment, (3) Organizational Readiness Assessment, and (4) Reconciliation. Each stage has clearly defined inputs, outputs and objectives to complete the 4-stage process.

- *Making the Agile Adoption Process Repeatable*

  By introducing four clearly defined stages and activities to help guide agile adoption efforts, the 4-Stage Process has explicitly defined portions of the agile adoption process. Once a process is defined, it can be repeated because different organizations can undergo the same process (agile adoption) knowing which activities and tasks need to be completed.

- *Introducing the notion of Pre-adoption Assessment*

  Stage 1 of the 4-Stage Process introduces the concept of pre-adoption assessment, which allows organizations to determine, before the adoption starts, aspects of the organization that could cause the adoption of an agile practice to fail. Conducting a pre-adoption assessment before starting the agile adoption effort can save an organization from needlessly spending time, money and effort on an SPI that has little chance of success.

- *Defining an Approach for Determining the Agile Potential of Individual Projects*

  Stage 2 and 3 in the 4-Stage Process provide the organization with two levels of assessment. The first is at a project level (stage 2) and the second is on an organizational level (stage 3). Stage 2, the Project Level Assessment, determines the target agile level (based on the SAMI) for the project after assessing certain organizational factors that are needed to support the adoption of certain agile practices. Stage 2 assesses only the organizational factors that cannot be changed or are outside the organization's control. The Organizational Readiness Assessment (stage 3), focuses on the organizational factors that can be changed and are within the organization's control. Hence, with Stage 2 and Stage 3 working together, the Agile Adoption Framework accommodates the uniqueness of each project and at the same time, recognizes that each project is surrounded by, and is a part of an overall organization that must be ready to adopt the requisite agile practices.

The Sidky Agile Measurement Index (SAMI): is a tailorable measurement index capable of measuring the agile potential of projects and organizations relative to the essential values and qualities of agility. This capability stems from the structure of the measurement index where it groups the agile practices in a synergistic manner based on the essential agile qualities and values they help introduce in the development process. The SAMI provides a number of contributions:

- *A unique approach to measuring agile potential*

  The SAMI has 4 main components: (1) Agile Levels, (2) Agile Principles, (3) Agile Practices, and (4) Indicators. The structure of the SAMI introduces an intuitive way for creating an agile measurement index. The "units" for the SAMI are the Agile Levels. Each level is populated with a set of Agile Practices which represent the basic measuring element of the index. The population process for each Agile Level is guided by the Agile Principles to ensure that each agile level manifests as many characteristics of an agile process as possible. The Indicators associated with each agile practice provide assessors with a quantifiable means to assess the readiness of the organization for each agile practice. The aggregated results of the indicator values indicate the agile potential of a project or organization.

- *Defines a Hierarchy of over 300 Indicators*

  To measure the readiness of an organization to adopt a particular agile practice, several different factors within the organization must be assessed. The SAMI provides a hierarchal organization of those factors in Readiness Assessment Tables (RAT). Each RAT contains the organizational factor that will be assessed, the objective of the assessment, the assessment method, and the list of indicators that are to be used for the assessment. Using this hierarchal structure, the SAMI is able to manage over 300 different indicators to assess an organization's readiness relative to 40 different agile practices.

- *Illustrating a Roadmap for Agility*

  By organizing the Agile Practices into Agile Levels, the SAMI also provides the agile community with a "roadmap" for those aspiring to move toward agility. The definition of Agile Levels in the SAMI encourages organizations to adopt the practices within the first Agile Level first, then move to the practices encompassed within subsequent levels of agility. The mere structure of the SAMI, especially the Agile Levels, provides organizations with a roadmap for their journey toward agility

- *Redefining When an Organization is Considered Agile*

  Many organizations ask when are they considered agile. The SAMI redefines the term agile from being a definite state to a spectrum. Based on SAMI, the level or degree of agility is dependent on the number of agile practices an organization has adopted. Moreover, the level of agility is dependent on the adoption of a certain group of agile practices that work together to help introduce a new value or quality into the development process. Therefore, due to the SAMI, there is now the notion of "level of agility" or "degree of agility" not just whether or not your organization is agile.

6.2. Future Work

We view the Agile Adoption Framework as an initial contribution towards answering the complex question of *how* to adopt agile practices. There is much room for the framework to mature and grow through further research. The following points are some possible areas for development of the Agile Adoption Framework.

- *Conducting a longitudinal study*

  For the purpose of this dissertation, the feedback obtained from the members of the agile community regarding the Agile Adoption Framework lead to the substantiation of the framework. Nevertheless, it is necessary to conduct a longitudinal study that gathers empirical evidence to further validate the framework and empirically prove the benefits yielded from using the Agile Adoption Framework.

- *Incorporating Business Value in the SAMI*

  The agile community is always concerned about the business value realized from each step of the development process. This leads us to ask: what are the business values realized from each step of the agile adoption process? Future work may include identifying the different business values associated with achieving each Agile Level. This addition to the SAMI is especially useful for executives that need to see what value adopting a new Agile level would yield to the organization before supporting the adoption process.

- *Enhancing the hierarchy of indicators*

  The indicator hierarchy has two aspects that are suitable for further research: the content and the structure.

  The content of the indicators (i.e. the actual questions) can be enhanced by conducting a study in which agile development environments are closely observed and analyzed. This study will help refine and add indicators to the Agile Adoption Framework in order to ensure accurate and realistic results when using the indicators for readiness assessment.

  As for the structure of the indicator hierarchy, we intend to explore the possible acyclic nature of the hierarchy. This is an important aspect because it can be used to help increase the overall efficiency of the assessment process. More specifically, by determining that the hierarchy of indicators is acyclic, positive correlations between the indicators can be identified. As a result, a smaller and more efficient set of indicators can be integrated into the readiness assessment process without affecting the quality of that assessment.

- *Automation of the portions of the Framework.*

  Future work may also include developing tools to automate the pre-adoption assessment process and assist with the evaluation of the results. For example, the indicators can be prepared on OpScan (Optical Scan) forms. Therefore, when an assessment is conducted in an organization, the developers, managers, customers and assessors are each given their respective OpScan forms. Once they answer the questions, the forms are automatically scanned and a program analyzes the results and suggests the set of agile practices most suitable for the organization to adopt.

6.3. Summary

In summary, we propose the Agile Adoption Framework as an approach to guide and assist organizations in their quest to adopt agile practices. Through identifying and assessing the presence of *discontinuing factors*, organizations can make a go/no-go decision regarding the move toward agility. By determining the *target level for a project* and then assessing the *organization* to determine the extent to which it is ready to achieve that target level of agility, the framework manages to provide coaches with a realistic set of agile practices for the project to adopt. The 4-Stage process assessment, through its utilization of the agile measurement index, provides an extensive outline of the areas within the organization that need improvement *before* the adoption effort starts.

While we recognize that the framework has yet to reach it full potential, we are encouraged by the comments given to the Agile Adoption Framework from members of the agile community:

- "*I think this is fantastic (work)*" –Agile consultant with 12 years experience
- "*This is the RIGHT time for this work! Excellent Job"* – Agile consultant with 8 years experience
- *"Overall this is first-class work and I endorse this work as legitimate in its interest and merit to our industry"* (paraphrased due to length*)* – XP Coach with 6 years experience.

7. References

[1] *Declaration of Interdependence*, http://pmdoi.org/, 2005.

[2] *Manifesto for Agile Software Devleopment*, www.agilemanifesto.org, Utah, Feb 2001.

[3] P. Abrahamsson, S. Outi, J. Ronkainen and J. Warsta, *Agile software development methods - Review and analysis*, VTT Electronics Finland, 2002, pp. 112.

[4] K. J. Aguanno, *Managing Agile Projects*, Multi-Media Publications Inc., 2005.

[5] S. Ambler, *Agile Modeling: Effective Practices for Extreme Programming and the Unified Process* Wiley, 2002.

[6] S. Ambler, *Communication on Agile Software Projects*, http://www.agilemodeling.com.

[7] S. W. Ambler and P. J. Sadalage, *Refactoring Databases: Evolutionary Database Design*, Addison-Wesley Professional, 2006.

[8] L. Amy and C. Raylene, *Effects of agile practices on social factors*, ACM Press, St. Louis, Missouri, 2005.

[9] J. Arthur and R. Nance, *Managing Software Quality: A Measurement Framework for Assessment and Prediction*, Springer, 2002.

[10] O. Balci, R. Adams, D. Myers and R. Nance, *A coollaborative Evaluation Enviornment for Credibility Assessment of Modeling and Simulation Applications Winter Simulation Conference*, 2002.

[11] L. Barnett, *Agile Survey Results: Solid Experience And Real Results Agile Journal*, 2006.

[12] L. Barnett and C. Schwaber, *Adopting Agile Development Processes; Improve Time-To-Benefits For Software Projects* Forrester Research, 2004.

[13] V. Basili, *Software modeling and measurement: the Goal/Question/Metric paradigm*, University of Maryland at College Park, 1992, pp. 24.

[14] V. R. Basili, *Software development: a paradigm for the future*, 1989, pp. 471-485.

[15] F. Bauer, *Software Engineering*, Information Processing 71 (1972).

[16] Beck, *Test Driven Development: By Example*, Addison-Wesley Longman Publishing Co., Inc., 2002.

[17] K. Beck, *Extreme Programming Explained: Embrace Change*, Addison-Wesley, 2000.

[18] K. Beck, R. C. Martin, A. Cockburn, M. Fowler and J. Highsmith, *Manifesto for Agile Software Devleopment*, www.agilemanifesto.org, Utah, Feb 2001.

[19] B. Boehm, *Get Ready for Agile Methods, with Care*, Computer, Vol. 35 (2002), pp. pp. 64-69

[20] B. Boehm and V. R. Basili, *Top 10 list [software development]*, Computer, 34 (2001), pp. 135-137.

[21] B. Boehm and R. Turner, *Management challenges to implementing agile processes in traditional development organizations*, Software, IEEE, 22 (2005), pp. 30-39.

[22] B. W. Boehm and R. Turner, *Balancing Agility and Discipline*, Addison-Wesley Professional, Boston, 2003.

[23] A. Cockburn, *Personal Communication* Salt Lake City, UT, November 2006.

[24] A. Cockburn, *Agile Software Development* Pearson Education, Indianapolis, 2001.

[25] A. Cockburn, *Agile Software Development: The Cooperative Game (2nd Edition) (Agile Software Development Series)*, Addison-Wesley Professional, 2006.

[26] A. Cockburn, *Crystal Clear: A Human-Powered Methodology for Small Teams* Addison-Wesley Professional, 2004.

[27] A. Cockburn and J. Highsmith, *Agile Software Development: The People Factor*, Computer, Volume 34 (2001), pp. Pages: 131 - 133

[28] M. Cohn, *Personal Communication*, Dayton, OH, October 2006.

[29] M. Cohn, *Agile Estimating and Planning* Prentice Hall PTR, 2005.

[30] M. Cohn, *User Stories Applied*, Addison -Wesley, Boston, 2004.

[31] M. Cohn and D. Ford, *Introducing an Agile Process to an Organization* Computer, Vol. 36 (2003), pp. 74-78.

[32] J. Collins, *Good to Great: Why Some Companies Make the Leap... and Others Don't*, Collins 2001.

[33] S. Datta, *Agility measurement index: a metric for the crossroads of software development methodologies*, *Proceedings of the 44th annual Southeast regional conference*, ACM Press, Melbourne, Florida, 2006.

[34] E. Derby and D. Larsen, *Agile Retrospectives: Making Good Teams Great*, Pragmatic Bookshelf, 2006.

[35] J. Drobka, D. Noftz and R. Rekha, *Piloting XP on four mission-critical projects*, Software, IEEE, 21 (2004), pp. 70-75.

[36] P. Duvall, S. Matyas and A. Glover, *Continuous Integration: Improving Software Quality and Reducing Risk* 2007.

[37] E. Eckman, *XP Transition Roadmap*, *Third International Conference on Programming and Agile Processes in Software Engineering (XP2002)*, Sardinia, Italy, 2002.

[38] A. Elssamadisy, *Personal Communication*, Amherst, MA, October 2006.

[39] A. Elssamadisy, *Getting Beyond "It Depends!" Being Specific But Not Prescriptive About Agile Practice Adoption Agile Journal*, 2006.

[40] K. E. Emam and N. H. Madhavji, *Elements of Software Assessment & Improvement*, IEEE Computer Society, USA, 1999.

[41] M. Fowler, K. Beck, J. Brant, W. Opdyke and D. Roberts, *Refactoring: Improving the Design of Existing Code*, Addison Wesley, Reading, Massachusetts, 1999.

[42] M. Gladwell, *The Tipping Point: How Little Things Can Make a Big Difference* Back Bay Books, 2002.

[43] R. B. Grady, *Successful software process improvement*, Prentice-Hall, Inc., 1997.

[44] P. Hersey and K. Blanchard, *Management of Organizational Behavior: Leading Human Resources* Prentice Hall 1982.

[45] J. Highsmith, *Agile Software Development Ecosystems*, Pearson Education, Indianapolis, 2002.

[46] J. Highsmith, *Agile: From Rogue Teams to Enterprise Acceptance Cutter Consortium: Business Technology Trends and Impacts*, 2006.

[47] G. Hofstede, *Cultures and Organizations, Software of the Mind: Intercultural Cooperation and its Importance for Survival* McGraw-Hill, 1996.

[48] W. S. Humphrey, *Introduction to the team software process*, Addison-Wesley Longman Ltd., 2000.

[49] W. S. Humphrey, *Managing technical people: innovation, teamwork, and the software process*, Addison-Wesley Longman Publishing Co., Inc., 1996.

[50] A. Hunt and D. Thomas, *Pragmatic Unit Testing in C\# with NUnit*, The Pragmatic Programmers, 2004.

[51] J. Hunt, *Agile software construction* Springer, London 2006.

[52] ISO/IEC, *ISO/IEC 15504-7 : Guide for use in Process Improvement*, *Information Technology - Software Process assessment - Part 7*, ISO/IEC, 1998, pp. 43.

[53] P.-H. Jan and J. Jorn, *AIM - Ability Improvement Model*, 2005.

[54] B. Jason, M. John, M. Erik, B. Matthew and A. Azeem, *Tailoring XP for Large System Mission Critical Software Development*, 2002.

[55] T. Kähkönen, *Life Cycle Model for Software Process Improvement Project Deploying an Agile Method*, *ICAM 2005 International Conference on Agility*, Helsinki, Finland, 2005.

[56] W. Kelly, P. Mary, M. Ron, S. Nancy Van and P. Bill, *Agile Methods for Safety-Critical Software Development*, 2004.

[57] A. S. Koch, *Agile software development : evaluating the methods for your organization*, Artech House, Boston, MA 2005.

[58] **S. Komi-Sirviö**, *Development and Evaluation of Software Process Improvement Methods*, VTT Publications 2004.

[59] S. Kuppuswami, K. Vivekanandan, P. Ramaswamy and P. Rodrigues, *The effects of individual XP practices on software development effort*, SIGSOFT Softw. Eng. Notes, 28 (2003), pp. 6-6.

[60] C. Larman, *Agile and Iterative Development*, Pearson Education, Boston, 2004.

[61] A. Law and R. Charron, *Effects of agile practices on social factors*, *Proceedings of the 2005 workshop on Human and social factors of software engineering*, ACM Press, St. Louis, Missouri, 2005.

[62] T. Little, *Context-adaptive agility: managing complexity and uncertainty*, Software, IEEE, 22 (2005), pp. 28-35.

[63] T. Little, F. Greene, T. Phillips, R. Pilger and R. Poldervaart, *Adaptive Agility* in I. Software, ed., *Agile Development Conference (ADC'04)*, IEEE Software, 2004, pp. pp. 63-70.

[64] R. C. Martin, *Agile Software Development, Principles, Patterns, and Practices*, Prentice Hall 2002.

[65] P. W. Mattessich, M. Murray-Close and B. Monsey, *Collaboration: What Makes It Work, 2nd Edition: A Review of Research Literature on Factors Influencing Successful Collaboration* Amherst H. Wilder Foundation, 2004.

[66] R. McFeeley, *IDEAL: A User's Guide for Software Process Improvement*, Software Engineering Institute (SEI), 1996, pp. 222.

[67] S. Nerur, R. Mahapatra and G. Mangalaraj, *Challenges of migrating to agile methodologies*, Commun. ACM, 48 (2005), pp. 72-78.

[68] J. W. Newkirk and R. C. Martin, *Extreme Programming in Practice* Prentice Hall, 2001.

[69] M. C. Paulk, *Agile Methodologies and Process Discipline*, CrossTalk: The Journal of Defense Software Engineering (2002).

[70] K. Peterson, *Collaboration: The Key to Enterprise Agile Adoption* Cutter IT Journal, July 05 (2005).

[71] A. Pukinskis, *5 stumbling blocks for new corporate agile projects*, *the agile blog*, 2005.

[72] D. Rosenberg, M. Stephens and M. Collins-Cope, *Agile development with ICONIX process : people, process, and pragmatism* Apress Berkeley, CA 2005.

[73] A. Rueping, *Agile Documentation : A Pattern Guide to Producing Lightweight Documents for Software Projects* John Wiley & Sons 2003.

[74] O. Salo, *Enabling Software Process Improvement in Agile Software Development Teams and Organisations* VTT Publications, 2006.

[75] A. Sanjiv, *Personal Communication*, Reston, VA, October 2006.

[76] B. Schatz and I. Abdelshafi, *Primavera gets agile: a successful transition to agile development*, Software, IEEE, 22 (2005), pp. 36-42.

[77] K. Schwaber and M. Beedle, *Agile Software Development with SCRUM*, Prentice Hall,, 2002.

[78] A. Sidky and J. D. Arthur, *Determining the Applicability of Agile Practices to Mission and Life-critical Systems*, *The 31st Annual Software Engineering Workshop (in conjunction with 3rd IEEE Systems and Software Week)*, Baltimore, Maryland, March 2007.

[79] M. K. Spayd, *Evolving agile in the enterprise: implementing XP on a grand scale*, 2003, pp. 60-70.

[80] J. Tabaka, *Collaboration Explained; Facilitation Skills for Software Project Leaders*, Addison-Wesley, 2005.

[81] W. Wake, *Personal Communication*, Richmond, VA, September 2006.

[82] W. C. Wake, *Extreme Programming Explored*, Addison-Wesley Professional, 2001.

[83] L. H. Werth, *Lecture Notes on Software Process Improvement*, Software Engineering Insitute 1993.

[84] L. Williams and R. Kessler, *Pair Programming Illuminated*, Addison-Wesley Longman Publishing Co., Inc., 2002.

[85] L. Williams, R. R. Kessler, W. Cunningham and R. Jeffries, *Strengthening the case for pair programming*, Software, IEEE, 17 (2000), pp. 19-25.

[86] I. Wilson and J. Wyer, *The Power of Collaborative Leadership:: Lessons for the Learning Organization* Butterworth-Heinemann, 2000.

Appendix A: Indicators

Indicators are essentially questions used by an assessor to measure certain characteristics of an organization or project, such as its developers, culture, management, software process, tools, project management practices, and physical environment.

A set of indicators, or questions, must accompany each agile practice or concept in the measurement index. The agile coach uses these indicators (or questions) to measure the extent to which

- discontinuing factors are present in an organization
- the organization is ready to adopt an agile practice or concept

Most indicators are based on a 5-point Likert summated scale, from 1 "strongly disagree" to 5 "strongly agree". A small number of indicators are based on other 5-point scales that are more appropriate to the characteristic being assessed.

The Agile Adoption Framework contains a set of over 300 suggested indicators, however the assessor performing the actual assessment may add other questions if needed.

Organization of the Indicators

The indicators used to assess each agile practice or discontinuing factor are grouped together in an *assessment table*. An assessment table captures the different organizational characteristics that need to be assessed for a practice and which indicator(s) should be used to assess each characteristic. The table below shows the assessment table used to assess the extent to which the organization is ready to adopt the agile practice, *Coding Standard*.

Category of Assessment	Area to be assessed	Characteristic to be assessed	To determine:	Assessment Method	Sample Indicators
People	Developers	Buy-In	Whether the developers see the benefit and are willing to apply coding standards	Interviewing	OR1_D21, OR1_D22
Process	Coding Standards	Existence	Whether there exists any kind of coding standards that are used	Observation	OR1_A2

Assessment table used to measure organizational readiness for an agile practice

The table above shows that in order to assess the organization's readiness for coding standards, the assessor first looks at the "people" in the organization with a particular focus on the developers. The characteristic that needs to be assessed with regards to the developers is their buy-in. The assessment of the developers' buy-in will help to determine whether or not the developers can recognize the benefits of coding standards and are willing to adopt them. The 5th column in the assessment table suggests a method of the assessment process; in this case it is interviewing. The last column contains the list of indicators that can be used for the assessment process. The assessor is free to use other assessment methods or indicators as long as the characteristics that need to be assessing are validly measured.

Each agile level in the agile measurement index contains a set of practices. Therefore, from an assessment viewpoint, each agile level will contain the assessment tables for all the practices encompassed with the level. For each level, all the indicators associated with the level are grouped together based on the group of people the question is targeted toward. This group is identified by the first letter after the underscore in the indicator's name. For example, the indicator OR1_**D**21 should be answered by a developer since the character after the underscore is a D. The codes used to denote the four different groups of people the indicators are targeted toward are:

- *A*: denotes an indicator that needs to be answered by the *assessor* (the one conducting the assessment)

- *D*: denotes an indicator that needs to be answered by a *developer* (anyone on the development side of the project)
- *M*: denotes an indicator that needs to be answered by a *manager* (anyone performing management related tasks with regards to the project)
- *C*: denotes an indicator that needs to be answered by a *customer executive* (a decision maker from the entity contracting to develop the software)

As for the letters before the underscore, they represent the stage within the process that usually uses the indicator. For example, OR1 implies that this indicator is used to measure the Organizational readiness for Agile Level 1. The list of all the codes used before the underscore and their meanings are:

- *DC*: represents indicators that used during *Stage 1: Identifying the Discontinuing Factors*
- *PL*: represents indicators that are used during *Stage 2: Project Level Assessment*
- *ORx:* represents indicators that are used during *Stage 3: Identifying the Organizational Readiness*, relative to *Agile level x*

Evaluation of the Indicators' Results

This section presents the evaluation methodology used after all the indicators are assessed. The sample assessment table below is used to explain the methodology.

Each discontinuing factor or agile practice uses an assessment table similar to the one below.

Category of Assessment	Area to be assessed	Characteristic assessed	To determine:	Assessment Method	Sample Indicators
People	Developers	Interaction	Whether there exists any levels of interaction between people hence laying a foundation for more team work	Interviewing	I1, I2, I3
		Collectivism	Whether people believe in group work and helping others or are they just concerned about themselves	Interviewing	I4
		Buy-In	Whether people are willing to work in teams	Interviewing	I5, I6
			Whether people recognize that their input is valuable in group work or not	Interviewing	I7, I8

After the assessor gathers the results from the surveys, he or she start a series of steps to evaluate the results. These steps and the evaluation methodology used are based off of the framework of the Evaluation Environment [10]. To use an automated tool to conduct an evaluation of the results, or to learn more about the evaluation environment please visit: http://www.orcacomputer.com/ee.

Step 1: Compute a weight for each indicator

The first step is to assign a weight to each indicator. A weight is a fractional value between 0 and 1 that expresses the indicator's level of influence on the parent factor or organizational characteristic. The weights of all the indicators belonging to the same factor must sum to 1. We will assume that all the indicators have an equal weight, however evaluators are free to assign indicators higher weights than other.

Therefore looking at the first factor in the example above, the weights can be computed as follows (under the assumption all indicators have an equal influence of the parent factor)

1 (sum of all weights) / **3** (number of indicators) = **0.334** (approximate weight per indicator)

Step 2: Compute weighed interval

After we computed the weight for each of the indicators, the next step is to compute the weighted intervals for each of the indicators. For the above example we will assume the following answers were given to the sample indicators of the first factor being assessed.

		Normalized Categories				
		V	W	X	Y	Z
Indicator	Sample Question	0%-15%	15%-40%	40%-60%	60%-85%	85%-100%
I1	Question 1	X				
I2	Question 2			X		
I3	Question 3					X

X Represents the answer that was chosen for that indicator

Once you have the answers from the sample indicators the next step is to multiple the weight of the indicator by the high and low end of the interval range selected for the indicator

Indicator Number	Computed Weight	Interval Low End	Interval High End	Interval Low End X Weight	Interval High End X Weight
I1	0.33334	0	15	0 X 0.3334 = **0**	15 X 0.3334 = **5**
I2	0.33334	40	60	40 X 0.3334 = **13**	60 X 0.3334 = **20**
I3	0.33334	85	100	85 X 0.3334 = **28**	100 X 0.3334 = **33**

Step 3: Calculate Result Range

The next step is to compute the Result Range by calculating the optimistic and pessimistic range for each factor. This is accomplished by summing up all the weighed intervals we got from the previous step. The example below highlights in more detail how this is done.

Pessimistic Result = Sum of all the weighed low end results from Step 2

Pessimistic Result: 0 + 13 +28 = 41

Optimistic Result = Sum of all the weighed high end results from Step 2

Optimistic Result: 5 +28 + 33 = 58

Your Result Range = 41 – 58

Step 4: Translate to Nominal Score

In many cases the assessment table indicates that several aspects need to be assessed in order to completely assess a certain characteristic of a factor. In those cases we do not compute and nominal value, rather we perform another round of aggregation, as demonstrated above, but on the next level up till we reach the level of the characteristic being assessed. In the assessment table presented earlier you can see that to assess Interaction and Collectivism we do not have to go through another cycle of aggregation. However to assess the buy-in we have to aggregate the indicators and then we have to aggregate the 2 different aspects of buy-in that we are assessing before we can move to the step that determines the nominal assessment result for the characteristic being assessed.

Once you have a result range for that a particular characteristic, and you are sure you do not have to perform more aggregation the next step is to map the result range to one of the nominal values presented below. These nominal values are the ones that are used to evaluate the fulfillment of a particular factor or not.

Not Achieved	**0%-35%**
Partially Achieved	**35%-65%**
Largely Achieved	**65%-85%**
Fully Achieved	**85% - 100%**

Nominal Values

If the Pessimistic - Optimistic (From Step 3) range fits within one of these intervals then that suffices, if they do not then obtain an average and then place that average in its necessary nominal range.

In our example the resultant score will be: **Partially Achieved**

Below is a sample of the evaluation template that would be used for the assessment table example given earlier

Category of Assessment	Area to be assessed	Characteristic assessed	**Nominal Value**	Weight	Low	High	Indicator	Weight	Low	High
Project Management	Developers	Interaction		1			I1	0.333		
							I2	0.333		
							I3	0.333		
		Collectivism		1			I4	1.000		
		Buy-In		0.5			I5	0.500		
							I6	0.500		
				0.5			I7	0.500		
							I8	0.500		

Evaluation Template for an Assessment Table

Once a nominal score is reached for each organizational characteristics being assessed, their nominal values are plugged in to the evaluation matrix similar to the table below to determine which areas need to be addressed before trying to adopt that particular agile practice.

Agile Practices for Agile Level 1	**Category of Assessment**	**Area to be assessed**	**Characteristic assessed**	**Not Achieved**	**Partially Achieved**	**Largely Achieved**	**Fully Achieved**
				0%-35%	35%-65%	65%-85%	85% - 100%
				Large Gap	Medium Gap	Small Gap	Minimal Gap
Collaborative Planning	People	Management	Management Style		X		
			Buy-In	X			
			Transparency			X	
		Developers	Power Distance				X
			Buy-In				X
	Project Management	Planning	Existence			X	
Collaborative Teams	Project Management	Developers	Interaction		X		
			Collectivism			X	
			Buy-In				X
Coding Standards	People	Developers	Buy-In				X
	Process	Coding Standards	Existence				X
Knowledge Sharing	People	Developers	Buy-In			X	
		Managers	Buy-In			X	
	Process	Knowledge Sharing	Availability	X			

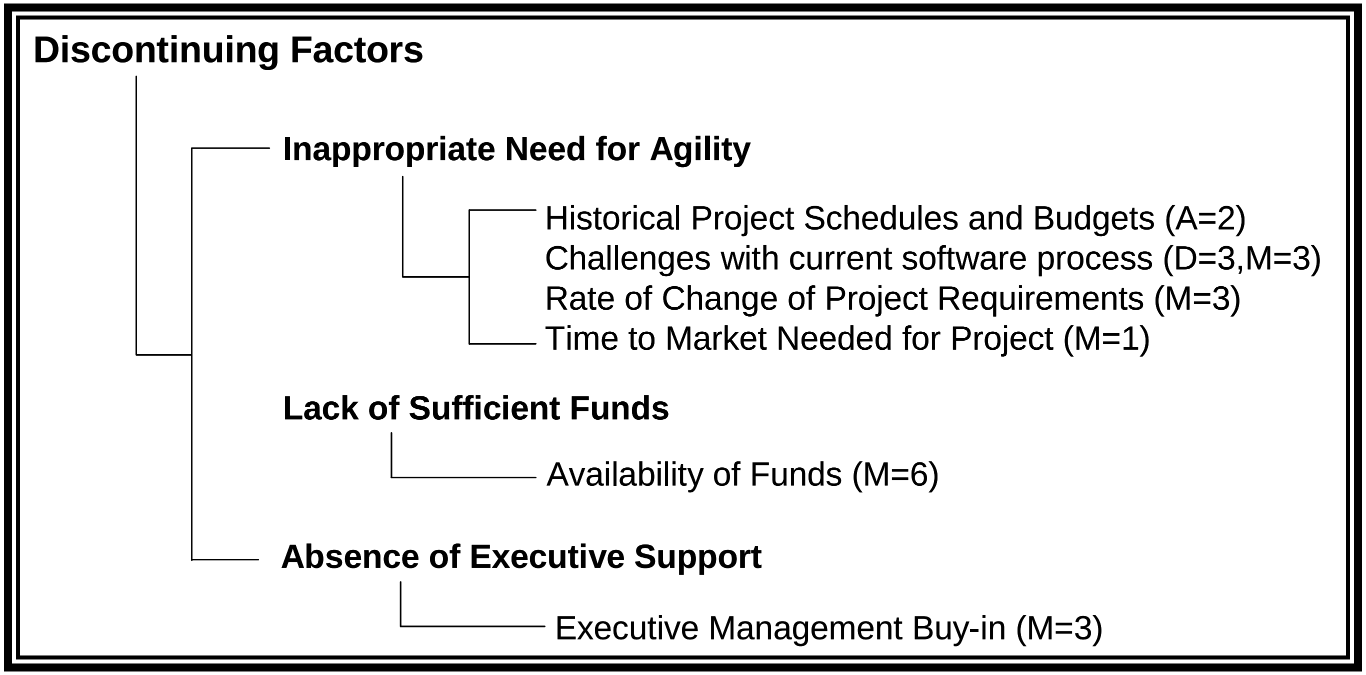
Sample Evaluation Matrix for Agile Level 1

The rest of this appendix presents the indicators contained in the Agile Adoption Framework. The indicators are grouped depending on which stage in the 4-Stage process they are used in. The first section in this appendix presents the indicators used to assess the discontinuing factors. The next section presents the indicators used to conduct a project level assessment (to identify the target level of agility for a project). Then the last sections include all the indicators used in Stage 3 to assess the organizational readiness for each individual agile practice.

Each section starts with an *indicator map* illustrating the hierarchy of indicators used during the associated stage. Following the indicator map are the assessment tables. Finally, the actually surveys where the indicators are found are grouped together based on who the indicator is addressed to (e.g. developers, managers, assessors, or customers).

Indicators related to
Stage 1: Discontinuing Factors

Indicator Map



Discontinuing Factors

Inappropriate Need for Agility

Historical Project Schedules and Budgets (A=2)
Challenges with current software process (D=3,M=3)
Rate of Change of Project Requirements (M=3)
Time to Market Needed for Project (M=1)

Lack of Sufficient Funds

Availability of Funds (M=6)

Absence of Executive Support

Executive Management Buy-in (M=3)

The number of within the parenthesis indicators denotes the number of indicators used to measure the related organizational characteristic. The letter preceding the number of indicators denotes who should provide the answer to the indicator's question.

Assessment Tables for Discontinuing Factors

Discontinuing Factor 1: Inappropriate Need for Agility

Category of Assessment	**Area to be assessed**	**Characteristic (s) to be assessed**	**To determine:**	**Assessment Method**	**Sample Indicators**
Organization	Project History	Schedule and Budget	Whether or not the organization has a history of having projects that go over time and budget	Observation	DC_A1, DC_A2
	Software Process	Problems	Whether or not the organization is facing any problems or dissatisfaction with the current software process	Interviewing	DC_D1, DC_D2, DC_D3, DC_M1, DC_M2, DC_M3
Project	Requirements	Rate of Change	Whether or not the project's requirements are clear and well defined, thus predicting no change, or whether or not the requirements need to be flexible and/or might change	Interviewing	DC_M5, DC_M6, DC_M7
	Delivery	Time to Market	Whether or not the project has to be developed quickly in order to introduce it to the market as soon as possible	Interviewing	DC_M4

Discontinuing Factor 2: Lack of Sufficient Funds

Category of Assessment	**Area to be assessed**	**Characteristic(s) to be assessed**	**To determine:**	**Assessment Method**	**Sample Indicators**
Organization	Budget	Availability of Funds	Whether or not the organization has funds to be spent on the adoption process of agile processes and is willing to spend them on the adoption process	Interviewing	DC_M10, DC_M11, DC_M12, DC_M13, DC_M14, DC_M15

Discontinuing Factor 3: Absence of Executive Support

Category of Assessment	**Area to be assessed**	**Characteristic(s) to be assessed**	**To determine:**	**Assessment Method**	**Sample Indicators**
People	Managers / Executives	Buy-in	Whether or not executive-level management can see benefits of adopting agile processes and will buy in to the development of agile software	Interviewing	DC_M3, DC_M8, DC_M9

The Surveys Encompassing the Indicators

Survey for Developers

	Statements	Nominal Values				
		V	W	X	Y	Z
DC_D1	There are many areas in the development process that always cause problems and/or are inefficient.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
DC_D2	The current development process is insufficient and/or does not produce good software.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
DC_D3	There is a need to change the software process in the organization.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree

Survey for Assessors

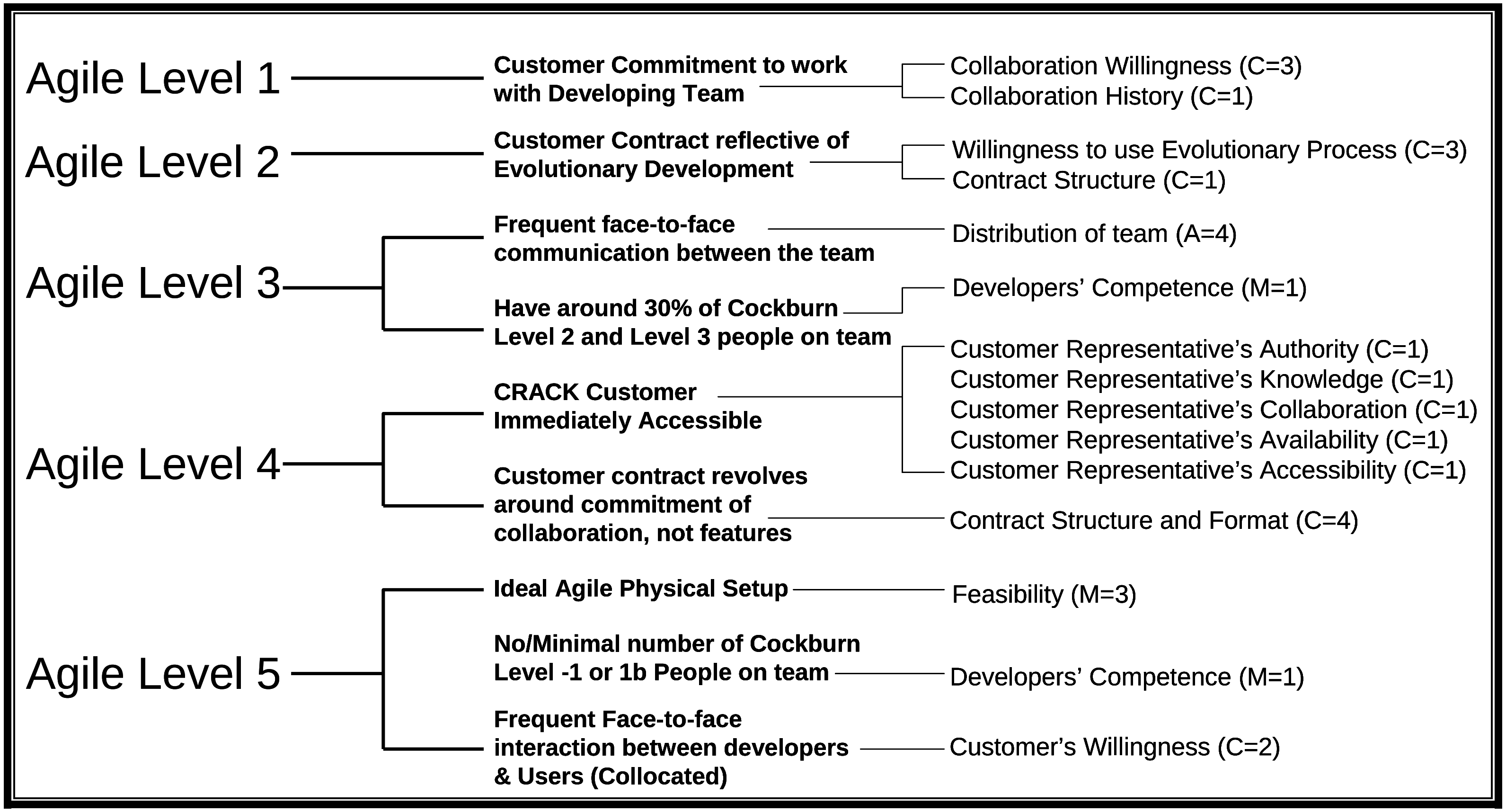
Indicator	Statements	Nominal Values				
		V	W	X	Y	Z
DC_A1	It can be concluded from the previous project plans and the project delivery documents that the organization has been on-time when delivering its projects.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
DC_A2	It can be concluded from previous project estimates and the project delivery documents that the organization has been within budget for its delivered projects.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree

Survey for Managers/Executives

Indicator	Statements	Nominal Values				
		V	W	X	Y	Z
DC_M1	There are some areas in the current development process that frequently cause problems and/or are inefficient.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
DC_M2	The current development process is insufficient and/or does not produce good software.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
DC_M3	There is a need to change the software process in the organization.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
DC_M4	The customer/client needs to introduce the product to the market quickly. (short time to market).	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
DC_M5	There is a high probability that requirements will change during the development process.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
DC_M6	Not all the requirements will be known before development starts for the project.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
DC_M7	The deliverables for this project are unknown.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
DC_M8	In general, employing agile processes help organizations overcome their software development challenges and/or respond better to customer requests.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
DC_M9	An Agile Development approach is ideal for the upcoming project.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
DC_M10	The organization has money allocated for training.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
DC_M11	The organization has money allocated for process improvement and/or organizational change.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
DC_M12	The organization is willing to spend on training people about Agile Processes.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
DC_M13	The organization is willing to spend whatever it takes for project to adopt an Agile Development approach.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
DC_M14	The organization has the necessary funds to undergo the process of adopting an agile development approach for the upcoming project.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
DC_M15	If adopting an agile process means buying new software, the organization is able and ready to spend on such software.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree

Indicators related to Stage 2: Project Level Assessment

Indicator Map


Agile Level 1
Customer Commitment to work with Developing Team
Collaboration Willingness (C=3)
Collaboration History (C=1)
Agile Level 2
Customer Contract reflective of Evolutionary Development
Willingness to use Evolutionary Process (C=3)
Contract Structure (C=1)
Agile Level 3
Frequent face-to-face communication between the team
Distribution of team (A=4)
Have around 30% of Cockburn Level 2 and Level 3 people on team
Developers' Competence (M=1)
Agile Level 4
CRACK Customer Immediately Accessible
Customer Representative's Authority (C=1)
Customer Representative's Knowledge (C=1)
Customer Representative's Collaboration (C=1)
Customer Representative's Availability (C=1)
Customer Representative's Accessibility (C=1)
Customer contract revolves around commitment of collaboration, not features
Contract Structure and Format (C=4)
Agile Level 5
Ideal Agile Physical Setup
Feasibility (M=3)
No/Minimal number of Cockburn Level -1 or 1b People on team
Developers' Competence (M=1)
Frequent Face-to-face interaction between developers & Users (Collocated)
Customer's Willingness (C=2)

The number of within the parenthesis indicators denotes the number of indicators used to measure the related organizational characteristic. The letter preceding the number of indicators denotes who should provide the answer to the indicator's question.

Assessment Tables for Limiting Agile Practices

Limiting Agile Practice within Agile Level 1

Customer Commitment to work with Developing Team

Category of Assessment	Area to be assessed	Characteristic(s) to be assessed	To determine:	Assessment Method	Sample Indicators
Customer	Collaboration	Willingness	Whether or not the customer is committed to work with Developing Team	Interviewing	PL_C1, PL_C3, PL_C4
		History	Whether or not this customer has collaborated with any development team before	Interviewing	PL_C2

Limiting Agile Practice within Agile Level 2

Customer Contract reflective of Evolutionary Development

Category of Assessment	Area to be assessed	Characteristic(s) to be assessed	To determine:	Assessment Method	Sample Indicators
Customer	Evolutionary Process	Willingness	Whether or not the customer agrees to an evolutionary development approach	Interviewing	PL_C5, PL_C6, PL_C8
	Contract	Structure	Whether or not the customer contract can be reflective of evolutionary development	Interviewing	PL_C7

Limiting Agile Practices within Agile Level 3

Frequent face-to-face communication between the team

Category of Assessment	Area to be assessed	Characteristic(s) to be assessed	To determine:	Assessment Method	Sample Indicators
People	Location	Distribution	Whether or not frequent face-to-face communication between team members is achievable	Observation	PL_A1, PL_A2, PL_A3, PL_A4

Have around 30% of Cockburn Level 2 and Level 3 people on team

Category of Assessment	Area to be assessed	Characteristic(s) to be assessed	To determine:	Assessment Method	Sample Indicators
People	Developers	Competence	Whether or not the development team has around 30% of Level 2 and Level 3 people on team	Interviewing	PL_M1

Limiting Agile Practices within Agile Level 4

Collaborative, Representative, Authorized, Committed, Knowledgeable (CRACK) Customer Immediately Accessible

Category of Assessment	Area to be assessed	Characteristic(s) to be assessed	To determine:	Assessment Method	Sample Indicators
Customer	Representative	Authority	Whether or not the customer representative has authority	Interviewing	PL_C9
		Knowledgeable	Whether or not the customer representative has detailed knowledge about the product	Interviewing	PL_C10
		Collaborative	Whether or not the customer representative is collaborative	Interviewing	PL_C11
		Availability	Whether or not the customer representative is available	Interviewing	PL_C12

		Accessibility	Whether or not the customer representative is immediately accessible	Interviewing	PL_C13

Customer contract revolves around commitment of collaboration, not features

Category of Assessment	Area to be assessed	Characteristic(s) to be assessed	To determine:	Assessment Method	Sample Indicators
Customer	Contract	Structure and format	Whether or not the customer's contract can revolve around commitment of collaboration, not features	Interviewing	PL_C14, PL_C15, PL_C16, PL_C17

Limiting Agile Practices within Agile Level 5

Ideal Agile Physical Setup

Category of Assessment	Area to be assessed	Characteristic(s) to be assessed	To determine:	Assessment Method	Sample Indicators
Organization	Physical Layout	Feasibility	Whether or not it is feasible to have an ideal agile physical setup	Interviewing	PL_M3, PL_M4, PL_M5

No/Minimal number of Cockburn Level -1 or 1b People on team

Category of Assessment	Area to be assessed	Characteristic(s) to be assessed	To determine:	Assessment Method	Sample Indicators
People	Developers	Competence	Whether or not no or a minimal number of Level -1 or 1b people exists on the development team	Interviewing	PL_M2

Frequent Face-to-face interaction between developers & Users (Collocated)

Category of Assessment	Area to be assessed	Characteristic(s) to be assessed	To determine:	Assessment Method	Sample Indicators
Customer	Face-to-Face Collaboration	Willingness	Whether or not the frequent face-to-face interaction between developers and customer is achievable	Interviewing	PL_C18, PL_C19

The Surveys Encompassing the Indicators

Survey for Customer Executives

Indicator	Statements	Nominal Values				
		V	W	X	Y	Z
PL_C1	I am willing to dedicate time to take an active role in this project.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
PL_C2	In the past, I have dedicated time to collaborate with the development team.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
PL_C3	I believe that the development team should make most of the effort and that the customer should have to do little other than check on the project's status and do a final acceptance.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
PL_C4	I am committed to working with the development team.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
PL_C5	I agree to have the system developed in an iterative/incremental fashion as opposed to the approach of a big delivery at the end of the contracted time.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
PL_C6	I am willing to sign a contract to start development of a product whose requirements cannot be known ahead of time with certainty.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
PL_C7	I am willing to change the contract structure to reflect an evolutionary development approach. Evolutionary development implies that the requirements, plan, estimates, and solution evolve or are refined over the course of the iterations, instead of being fully defined and "frozen" in a major upfront specification effort before the development begins.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
PL_C8	I am willing to accept an overall project plan and a detailed plan of the next iteration only. The customer does not have a problem with not receiving a GANTT or PERT chart of the whole project upfront.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
PL_C9	The customer representative(s) interacting with the contracted organization is (are) authorized to make decisions on the spot regarding the product specifications	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree

PL_C10	The customer representative(s) interacting with the contracted organization is (are) knowledgeable about the product domain (i.e. he/she is a domain expert or subject matter expert).	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
PL_C11	The customer representative(s) interacting with the contracted organization is (are) representative of the product's actual users.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
PL_C12	The customer representative is available for the development team to contact if it needs his/her input on something.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
PL_C13	A customer representative is immediately accessible to the development team if needed.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
PL_C14	I am willing to sign a contract that does not have a detailed enumeration of features and functions but broad goals and the success criteria. This allows the customer more flexibility to change and add requirements through out the development process.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
PL_C15	I am willing to accept a contract in which the time and budget, but not the features to be delivered, are fixed.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
PL_C16	I am willing to accept a contract that commits both sides to a degree of interaction and collaboration instead of a set of detailed requirements.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
PL_C17	I am willing to change its typical contract structure to reflect a new agile development approach. An agile development approach will give the customer the flexibility to change its requirements throughout the development process, and will deliver software earlier and in increments.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
PL_C18	The customer will be available for frequent face-to-face interaction with the development team.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
PL_C19	The customer is willing to be collocated with the development team.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree

Survey for Managers/Executives

Indicator	Statements	Nominal Values				
		V	W	X	Y	Z
PL_M1	What percentage of the full-time staff is of Cockburn Level 2 or Level 3 experts	0-5%	5-10%	10-15%	15-30%	30% or higher
PL_M2	Indicate the percentage of full-time staff who are Cockburn Level 2 or Level 3 experts.	30% or higher	15-30%	10-15%	5-10%	0-5%
PL_M3	The organization can have all the development personnel in a common room rather than separate offices or cubicles.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
PL_M4	The organization can set up the development rooms to better support agile development (furniture away from the walls).	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree

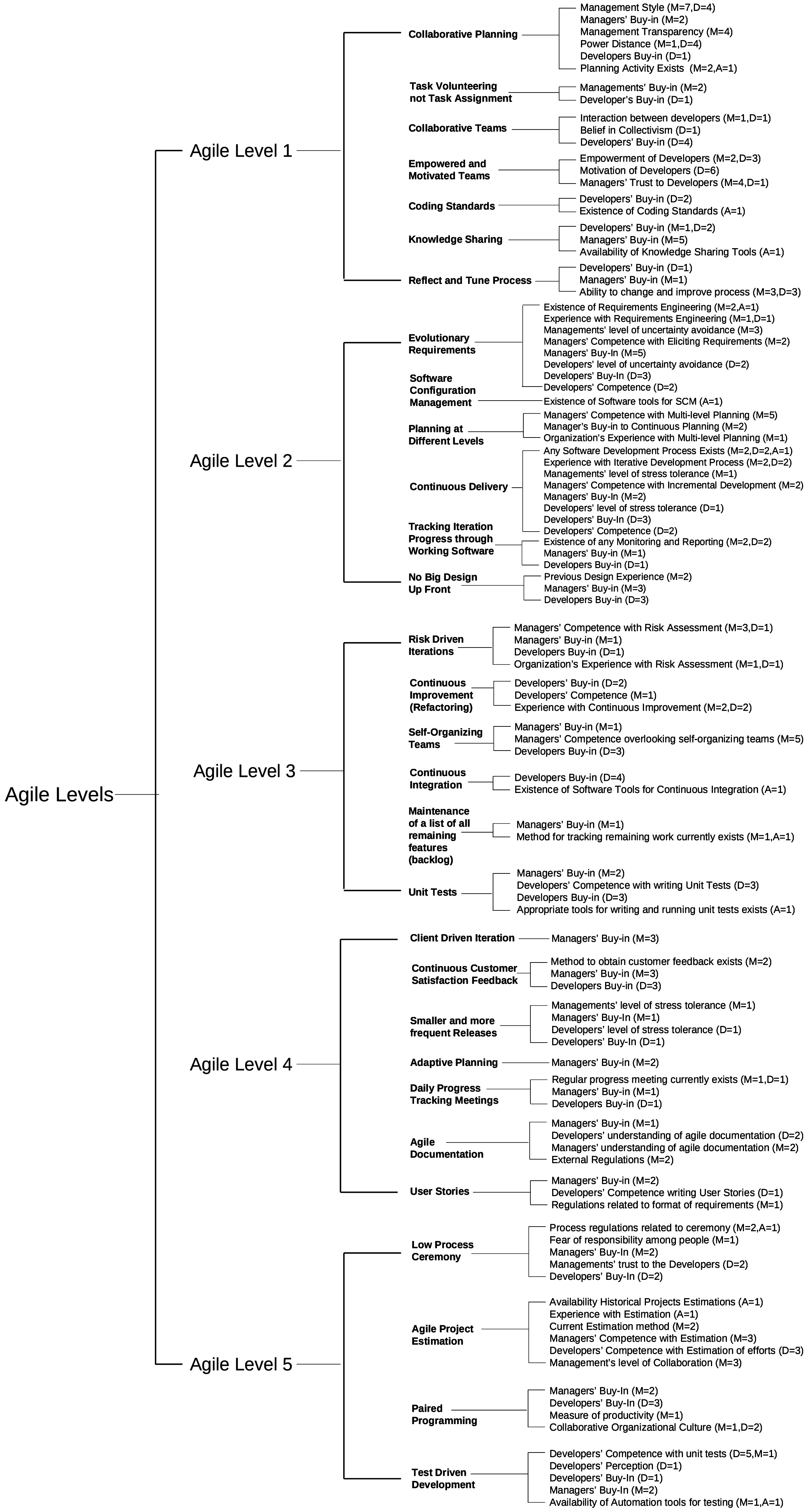
PL_M5	The organization can setup an environment where as much project information as possible is displayed on the walls (via whiteboards, cling sheets, or projectors).	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree

Survey for Assessors

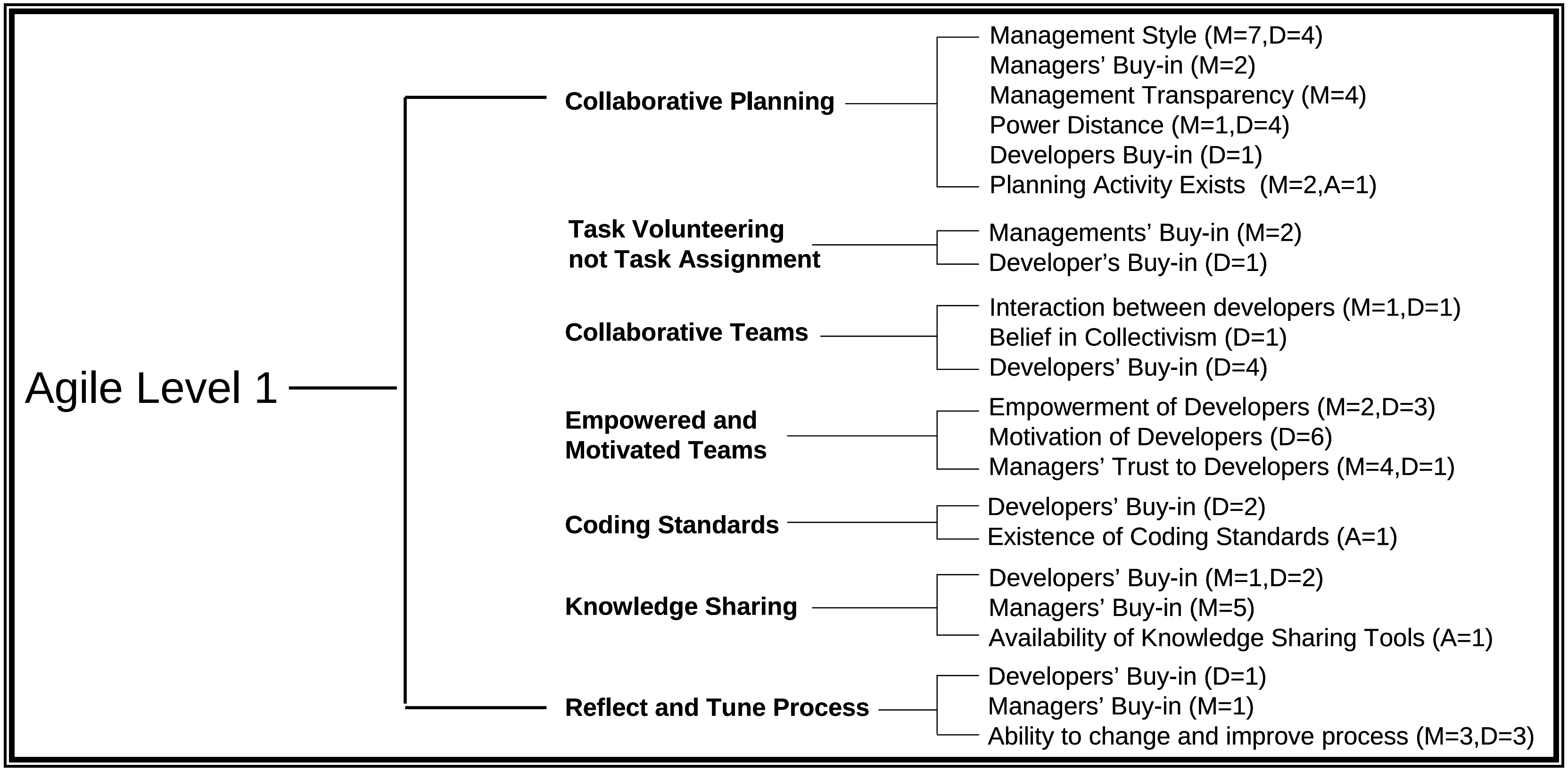
Indicator	**Statements**	Nominal Values				
		V	W	X	Y	Z
PL_A1	The development team is located where members can have frequent face-to-face communication.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
PL_A2	The geographic distribution of the development team can be best described as…	Within Flying Distance	Long distance driving	Within Daily Driving Distance	Within the same building	In the same room
PL_A3	Logistically, the development team can meet face-to-face.	Yearly or never	Monthly	Weekly	Daily	Hourly
PL_A4	It is likely for the development team to have frequent face-to-face communication.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree

Indicators related to
Stage 3: Organizational Readiness Assessment

For best view, please view this page using a portrait orientation


Agile Levels
Agile Level 1
Collaborative Planning
Management Style (M=7,D=4)
Managers' Buy-in (M=2)
Management Transparency (M=4)
Power Distance (M=1,D=4)
Developers Buy-in (D=1)
Planning Activity Exists (M=2,A=1)
Task Volunteering not Task Assignment
Managements' Buy-in (M=2)
Developer's Buy-in (D=1)
Collaborative Teams
Interaction between developers (M=1,D=1)
Belief in Collectivism (D=1)
Developers' Buy-in (D=4)
Empowered and Motivated Teams
Empowerment of Developers (M=2,D=3)
Motivation of Developers (D=6)
Managers' Trust to Developers (M=4,D=1)
Coding Standards
Developers' Buy-in (D=2)
Existence of Coding Standards (A=1)
Knowledge Sharing
Developers' Buy-in (M=1,D=2)
Managers' Buy-in (M=5)
Availability of Knowledge Sharing Tools (A=1)
Reflect and Tune Process
Developers' Buy-in (D=1)
Managers' Buy-in (M=1)
Ability to change and improve process (M=3,D=3)
Agile Level 2
Evolutionary Requirements
Existence of Requirements Engineering (M=2,A=1)
Experience with Requirements Engineering (M=1,D=1)
Managements' level of uncertainty avoidance (M=3)
Managers' Competence with Eliciting Requirements (M=2)
Managers' Buy-In (M=5)
Developers' level of uncertainty avoidance (D=2)
Developers' Buy-In (D=3)
Developers' Competence (D=2)
Software Configuration Management
Existence of Software tools for SCM (A=1)
Planning at Different Levels
Managers' Competence with Multi-level Planning (M=5)
Manager's Buy-in to Continuous Planning (M=2)
Organization's Experience with Multi-level Planning (M=1)
Continuous Delivery
Any Software Development Process Exists (M=2,D=2,A=1)
Experience with Iterative Development Process (M=2,D=2)
Managements' level of stress tolerance (M=1)
Managers' Competence with Incremental Development (M=2)
Managers' Buy-In (M=2)
Developers' level of stress tolerance (D=1)
Developers' Buy-In (D=3)
Developers' Competence (D=2)
Tracking Iteration Progress through Working Software
Existence of any Monitoring and Reporting (M=2,D=2)
Managers' Buy-in (M=1)
Developers Buy-in (D=1)
No Big Design Up Front
Previous Design Experience (M=2)
Managers' Buy-in (M=3)
Developers Buy-in (D=3)
Agile Level 3
Risk Driven Iterations
Managers' Competence with Risk Assessment (M=3,D=1)
Managers' Buy-in (M=1)
Developers Buy-in (D=1)
Organization's Experience with Risk Assessment (M=1,D=1)
Continuous Improvement (Refactoring)
Developers' Buy-in (D=2)
Developers' Competence (M=1)
Experience with Continuous Improvement (M=2,D=2)
Self-Organizing Teams
Managers' Buy-in (M=1)
Managers' Competence overlooking self-organizing teams (M=5)
Developers Buy-in (D=3)
Continuous Integration
Developers Buy-in (D=4)
Existence of Software Tools for Continuous Integration (A=1)
Maintenance of a list of all remaining features (backlog)
Managers' Buy-in (M=1)
Method for tracking remaining work currently exists (M=1,A=1)
Unit Tests
Managers' Buy-in (M=2)
Developers' Competence with writing Unit Tests (D=3)
Developers Buy-in (D=3)
Appropriate tools for writing and running unit tests exists (A=1)
Agile Level 4
Client Driven Iteration
Managers' Buy-in (M=3)
Continuous Customer Satisfaction Feedback
Method to obtain customer feedback exists (M=2)
Managers' Buy-in (M=3)
Developers Buy-in (D=3)
Smaller and more frequent Releases
Managements' level of stress tolerance (M=1)
Managers' Buy-In (M=1)
Developers' level of stress tolerance (D=1)
Developers' Buy-In (D=1)
Adaptive Planning
Managers' Buy-in (M=2)
Daily Progress Tracking Meetings
Regular progress meeting currently exists (M=1,D=1)
Managers' Buy-in (M=1)
Developers Buy-in (D=1)
Agile Documentation
Managers' Buy-in (M=1)
Developers' understanding of agile documentation (D=2)
Managers' understanding of agile documentation (M=2)
External Regulations (M=2)
User Stories
Managers' Buy-in (M=2)
Developers' Competence writing User Stories (D=1)
Regulations related to format of requirements (M=1)
Agile Level 5
Low Process Ceremony
Process regulations related to ceremony (M=2,A=1)
Fear of responsibility among people (M=1)
Managers' Buy-In (M=2)
Managements' trust to the Developers (D=2)
Developers' Buy-In (D=2)
Agile Project Estimation
Availability Historical Projects Estimations (A=1)
Experience with Estimation (A=1)
Current Estimation method (M=2)
Managers' Competence with Estimation (M=3)
Developers' Competence with Estimation of efforts (D=3)
Management's level of Collaboration (M=3)
Paired Programming
Managers' Buy-In (M=2)
Developers' Buy-In (D=3)
Measure of productivity (M=1)
Collaborative Organizational Culture (M=1,D=2)
Test Driven Development
Developers' Competence with unit tests (D=5,M=1)
Developers' Perception (D=1)
Developers' Buy-In (D=1)
Managers' Buy-In (M=2)
Availability of Automation tools for testing (M=1,A=1)

Indicator Map for Agile Level 1


Agile Level 1
Collaborative Planning
Management Style (M=7,D=4)
Managers’ Buy-in (M=2)
Management Transparency (M=4)
Power Distance (M=1,D=4)
Developers Buy-in (D=1)
Planning Activity Exists (M=2,A=1)
Task Volunteering not Task Assignment
Managements’ Buy-in (M=2)
Developer’s Buy-in (D=1)
Collaborative Teams
Interaction between developers (M=1,D=1)
Belief in Collectivism (D=1)
Developers’ Buy-in (D=4)
Empowered and Motivated Teams
Empowerment of Developers (M=2,D=3)
Motivation of Developers (D=6)
Managers’ Trust to Developers (M=4,D=1)
Coding Standards
Developers’ Buy-in (D=2)
Existence of Coding Standards (A=1)
Knowledge Sharing
Developers’ Buy-in (M=1,D=2)
Managers’ Buy-in (M=5)
Availability of Knowledge Sharing Tools (A=1)
Reflect and Tune Process
Developers’ Buy-in (D=1)
Managers’ Buy-in (M=1)
Ability to change and improve process (M=3,D=3)

The number of within the parenthesis indicators denotes the number of indicators used to measure the related organizational characteristic. The letter preceding the number of indicators denotes who should provide the answer to the indicator’s question.

Assessment Tables for Agile Practices within Agile Level 1

Collaborative Planning (*Customers, Developers and Management plan together)*

Category of Assessment	Area to be assessed	Characteristic(s) to be assessed	To determine:	Assessment Method	Sample Indicators
People	Management	Management Style	Whether or not a collaborative or a command-control relation exists between managers and subordinates. The management style is an indication of whether or not management trusts the developers and vice-versa.	Interviewing	OR1_M1, OR1_M2, OR1_M3, OR1_M4, OR1_M5, OR1_M14, OR1_M17, OR1_D1 OR1_D2, OR1_D3, OR1_D4,
		Buy-In	Whether or not management is supportive of or resistive to having a collaborative environment	Interviewing	OR1_M9, OR1_M10,
		Transparency	Whether or not management can be open with customers and developers – No politics and secrets	Interviewing	OR1_M6, OR1_M7, OR1_M8, OR1_M13
	Developers	Power Distance	Whether or not people are intimidated/afraid to give honest feedback and participation in the presence of their managers	Interviewing	OR1_M11, OR1_D6, OR1_D7, OR1_D8, OR1_D9
		Buy-In	Whether or not the developers are willing to plan in a collaborative environment	Interviewing	OR1_D5
Project Management	Planning	Existence	Whether or not the organization does basic planning for its projects	Observation	OR1_A1
				Interviewing	OR1_M16, OR1_M18

Task Volunteering not Task Assignment

Category of Assessment	Area to be assessed	Characteristic(s) to be assessed	To determine:	Assessment Method	Sample Indicators
People	Management	Buy-In	Whether or not management will be willing to buy into and can see benefits from employees volunteering for tasks instead of being assigned	Interviewing	OR1_M12, OR1_M15
	Developers	Buy-In	Whether or not developers are willing to see the benefits from volunteering for tasks	Interviewing	OR1_D10

Collaborative Teams

Category of Assessment	Area to be assessed	Characteristic(s) to be assessed	To determine:	Assessment Method	Sample Indicators
People	Developers	Interaction	Whether or not any levels of interaction exist between people thus laying a foundation for more team work	Interviewing	OR1_M1, OR1_D15
		Collectivism	Whether or not people believe in group work and helping others or are just concerned about themselves	Interviewing	OR1_D16
		Buy-In	Whether or not people are willing to work in teams	Interviewing	OR1_D12, OR1_D11
			Whether or not people recognize that their input is valuable in group work	Interviewing	OR1_D23, OR1_D13

Empowered and Motivated Teams

Category of Assessment	Area to be assessed	Characteristic(s) to be assessed	To determine:	Assessment Method	Sample Indicators
People	Developers	Decision Making	Whether or not management empowers teams with decision making authority	Interviewing	OR1_M3, OR1_D4, OR1_D14, OR1_D17, OR1_M14
		Motivation	Whether or not people are treated in a way that motivates them	Interviewing	OR1_D14, OR1_D13, OR1_D23, OR1_D24, OR1_D25, OR1_D15
	Managers	Trust	Whether or not managers trust and believe in the technical team in order to truly empower them	Interviewing	OR1_M13, OR1_M14, OR1_M6, OR1_M12, OR1_D2

Coding Standards

Category of Assessment	Area to be assessed	Characteristic(s) to be assessed	To determine:	Assessment Method	Sample Indicators
People	Developers	Buy-In	Whether or not the developers see the benefit and are willing to apply coding standards	Interviewing	OR1_D21, OR1_D22
Process	Coding Standards	Existence	Whether or not any kind of coding standards exists that are used	Observation	OR1_A2

Knowledge Sharing

Category of Assessment	Area to be assessed	Characteristic(s) to be assessed	To determine:	Assessment Method	Sample Indicators
People	Developers	Buy-In	Whether or not developers believe in and can see the benefits of having project information communicated to the whole team	Interviewing	OR1_D18, OR1_D19, OR1_M19
	Managers	Buy-In	Whether or not managers believe in and can see the benefits of having project information communicated to the whole team	Interviewing	OR1_M6, OR1_M7, OR1_M20, OR1_M21, OR1_M22
Tools	Knowledge Sharing	Availability	Whether or not knowledge sharing tools are available and accessible (Wikis, Blogs …etc.)	Observation	OR1_A3

Reflect and Tune Process

Category of Assessment	Area to be assessed	Characteristic(s) to be assessed	To determine:	Assessment Method	Sample Indicators
People	Developers	Buy-in	Whether or not developers are willing to commit to reflecting about and tuning the process after each iteration or release	Interviewing	OR1_D26
	Managers	Buy-in	Whether or not management is willing to commit to reflecting about and tuning the process after each iteration or release	Interviewing	OR1_M23
Process	Process improvement	Capability	Whether or not the organization can handle process change in the middle of the project	Interviewing	OR1_D27, OR1_D28, OR1_D29, OR1_M24, OR1_M25, OR1_M26

The Surveys Encompassing the Indicators

Survey for Developers

Indicator	Statements	Nominal Values				
		V	W	X	Y	Z
OR1_D1	Your manager listens to your opinions regarding technical issues	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR1_D2	Your manager does not micro-manage you or your work.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR1_D3	Your manager encourages you to be creative and does not dictate to you what to do exactly.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR1_D4	Your manager gives you the authority to make decisions without referring back to him/her.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR1_D5	You would like to participate in the planning process of the project you will work on.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR1_D6	If your manager said or did something wrong, it is acceptable for you to correct and/or constructively criticize him/her face to face.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR1_D7	It is acceptable for you to express disagreement with your manager(s) without fearing their retribution.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR1_D8	In a group meeting, the customer suggested something about the product. You disagree and have a better idea; it is acceptable for you to express disagreement with your customer and suggest something better.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR1_D9	Other peoples' titles and positions intimidate people in the organization.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR1_D10	You would do a better job choosing your own task on a project instead of being assigned one by your manager.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR1_D11	You prefer working in a group.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR1_D12	Indicate how often you work in groups.	Never	Seldom	Sometimes	Usually	Always
OR1_D13	When in a group, you feel that your participation is important.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree

OR1_D14	Your manager seeks your input on technical issues.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR1_D15	Your team members seek your input on technical issues.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR1_D16	When you run into technical problems, you usually ask your team members about the solution.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR1_D17	You usually participate in the planning process of the project you are working on.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR1_D18	Project information should be communicated to the whole team.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR1_D19	There should be a mechanism for persistent knowledge sharing between team members.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR1_D20	People should use a wiki or a blog for knowledge sharing.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR1_D21	There should exist a coding standard for development.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR1_D22	If the organization has a coding standard, then developers should use it when coding, even in crunch time.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR1_D23	The organization values you and your expertise.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR1_D24	Your manager has high expectations of you.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR1_D25	You are motivated by your job.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR1_D26	You are willing to dedicate time after each iteration/release to review how the process could be improved.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR1_D27	You are willing to undergo a process change even if it requires some reworking of already completed work products.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR1_D28	If there is a need for process change, that change should not be considered a burden on the team even if significant process changes have been made previously during the project.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR1_D29	Process change in the middle of the project should not be considered a disruption since the process change is worth the benefit it will bring.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree

Survey for Managers/Executives

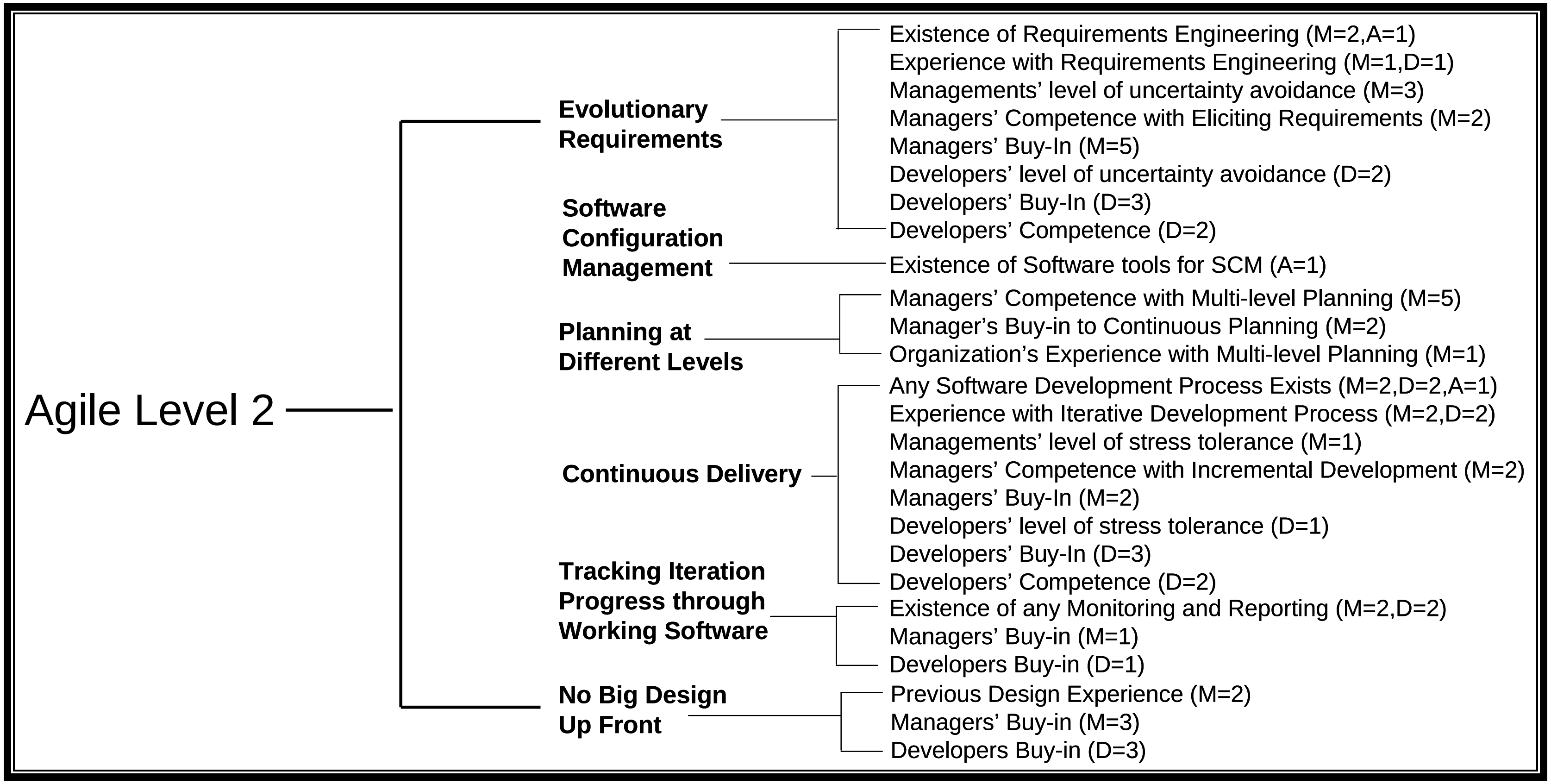
Indicator	Statements	Nominal Values				
		V	W	X	Y	Z
OR1_M1	You actively encourage interaction among your subordinates.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR1_M2	Irrelevant of your personal preferences, you encourage team work over individual work.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR1_M3	You usually seek your subordinates' opinions before making a decision.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR1_M4	You frequently brainstorm with your subordinates.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR1_M5	You frequently encourage your subordinates to find creative solutions to problems.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR1_M6	It is important for you to share project management information with your subordinates.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR1_M7	If you are needed and unreachable, at any point in time your subordinates have enough information to update the customer about the exact status of the project.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR1_M8	If a problem occurs that may affect the schedule or requirements of a project, you would update your client right away.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR1_M9	Developers should aid in the planning of a project.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR1_M10	Customers should be part of the planning of a project.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR1_M11	Other peoples' titles and positions intimidate people in the organization.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR1_M12	You allow your subordinates to choose their own tasks for a project	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR1_M13	Your subordinates have unregulated access to the customer.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR1_M14	You frequently seek the input of your subordinates on technical issues.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree

OR1_M15	You believe that subordinates would perform better and be more effective if they were to choose their own tasks.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR1_M16	You always create plans for a software development project.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR1_M17	It is important to involve other people while preparing the project plan.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR1_M18	The project plans are always documented.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR1_M19	When you prepare a project plan, it should not include the details of the project from start to end; it should be focused on the next iteration while giving an overview of the overall work	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR1_M20	Project information should be communicated to the whole team.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR1_M21	There should be a mechanism for persistent knowledge sharing between team members.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR1_M22	If there was a wiki or a blog set up for knowledge sharing, you believe people would use it.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR1_M23	You are willing to dedicate time after each iteration/release to review how the process could be improved.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR1_M24	You are willing to undergo a process change even if it requires some reworking of already completed work products.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR1_M25	If there is a need for process change, that change should not be considered a burden on the team even if significant process changes have been made previously during the project.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR1_M26	Process change in the middle of the project should not be considered a disruption since the process change is worth the benefit it will bring.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree

Survey for Assessors

Indicator	Statements	Nominal Values				
		V	W	X	Y	Z
OR1_A1	Old project documents show that previous projects have a project plan.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR1_A2	A review of documents or other information shows you that the organization has a coding standard.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR1_A3	A review of the tools available for use by the developers shows you that the organization has and uses knowledge sharing tools (Wikis, Blogs …etc.).	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree

Indicator Map for Agile Level 2


Agile Level 2
Evolutionary Requirements
Existence of Requirements Engineering (M=2,A=1)
Experience with Requirements Engineering (M=1,D=1)
Managements' level of uncertainty avoidance (M=3)
Managers' Competence with Eliciting Requirements (M=2)
Managers' Buy-In (M=5)
Developers' level of uncertainty avoidance (D=2)
Developers' Buy-In (D=3)
Developers' Competence (D=2)
Software Configuration Management
Existence of Software tools for SCM (A=1)
Planning at Different Levels
Managers' Competence with Multi-level Planning (M=5)
Manager's Buy-in to Continuous Planning (M=2)
Organization's Experience with Multi-level Planning (M=1)
Continuous Delivery
Any Software Development Process Exists (M=2,D=2,A=1)
Experience with Iterative Development Process (M=2,D=2)
Managements' level of stress tolerance (M=1)
Managers' Competence with Incremental Development (M=2)
Managers' Buy-In (M=2)
Developers' level of stress tolerance (D=1)
Developers' Buy-In (D=3)
Developers' Competence (D=2)
Tracking Iteration Progress through Working Software
Existence of any Monitoring and Reporting (M=2,D=2)
Managers' Buy-in (M=1)
Developers Buy-in (D=1)
No Big Design Up Front
Previous Design Experience (M=2)
Managers' Buy-in (M=3)
Developers Buy-in (D=3)

The number of within the parenthesis indicators denotes the number of indicators used to measure the related organizational characteristic. The letter preceding the number of indicators denotes who should provide the answer to the indicator’s question.

Assessment Tables for Agile Practices within Agile Level 2

Evolutionary Requirements

Category of Assessment	Area to be assessed	Characteristic(s) to be assessed	To determine:	Assessment Method	Sample Indicators
Process	Requirements Engineering	Existence	Whether or not the organization has an institutionalized procedure to gather requirements from its clients	Observation	OR2_A1
				Interviewing	OR2_M, OR2_M2
		Experience	Whether or not the organization has developed projects using the evolutionary requirements	Interviewing	OR2_D1, OR2_M3
People	Management	Uncertainty Avoidance	Whether or not management accepts and is comfortable with the uncertainty involved with deciding on requirements and features as late as possible	Interviewing	OR2_M4, OR2_M5, OR2_M6
		Competence	Whether or not the managers can recognize high-level (architecturally influential) requirements and differentiate them from detail requirements	Interviewing	OR2_M7, OR2_M8
		Buy-In	Whether or not management is willing to accept changes from the customer and that all changes are reversible	Interviewing	OR2_M6, OR2_M9, OR2_M0
			Whether or not management is willing to try evolutionary requirements over big upfront requirements gathering	Interviewing	OR2_M1, OR2_M2
	Developers	Uncertainty Avoidance	Whether or not the developers accept and are comfortable with the uncertainty involved with deciding on requirements and features as late as possible	Interviewing	OR2_D2, OR2_D3
		Buy-In	Whether or not the developers are willing to accept changes from the customer and that all changes are reversible	Interviewing	OR2_D4, OR2_D7, OR2_D8
		Competence	Whether or not the developers can recognize high-level (architecturally influential) requirements and differentiate them from detail requirements	Interviewing	OR2_D5,OR2_D6

Software Configuration Management

Category of Assessment	Area to be assessed	Characteristic(s) to be assessed	To determine:	Assessment Method	Sample Indicators
Environment	Software Tools	Existence	Whether or not the organization has tools for software configuration management	Observation	OR2_A3

Continuous Delivery (Incremental-Iterative development)

Category of Assessment	Area to be assessed	Characteristic(s) to be assessed	To determine:	Assessment Method	Sample Indicators
Process	Process Definition	Existence	Whether or not the organization has any process in place for development and is not relying on haphazard and ad-hoc approaches to software development	Observation	OR2_A2
				Interviewing	OR2_D9, OR2_D10, OR2_M13, OR2_M14
	Lifecycle	Experience	Whether or not the organization has previously used an incremental – iterative approach for developing systems	Interviewing	OR2_M15, OR2_M16 OR2_D11, OR2_D12
People	Management	Buy-In	Whether or not management will be willing to use an iterative-incremental development approach	Interviewing	OR2_M17, OR2_M18
		Stress	Whether or not managers can handle the additional stress of overseeing the delivery of workable iterations every 1-4 weeks	Interviewing	OR2_M19
		Competence	Whether or not the managers understand the principles of incremental-iterative development	Interviewing	OR2_M20, OR2_M21
	Developers	Stress	Whether or not the developers can handle the stress of delivering a workable iteration every 1-4 weeks	Interviewing	D_15
		Buy-In	Whether or not developers will be willing to use an iterative-incremental development approach	Interviewing	OR2_D13, OR2_D14, OR2_D18
		Competence	Whether or not the developers understand the principles of incremental-iterative development	Interviewing	OR2_D16, OR2_D17

Planning at different levels

Category of Assessment	Area to be assessed	Characteristic(s) to be assessed	To determine:	Assessment Method	Sample Indicators
People	Managers	Competence	Whether or not management understands the principles and significance of multi-level planning	Interviewing	OR2_M22, OR2_M23, OR2_M24, OR2_M25,

					OR2_M26
		Buy-in	Whether or not management is willing to commit to the process of continuously planning versus developing a one-time plan	Interviewing	OR2_M27, OR2_M28
Process	Planning	Experience	Whether or not the organization is experienced with multi-level or not	Interviewing	OR2_M29

Tracking Iteration Progress through Working Software

Category of Assessment	Area to be assessed	Characteristic(s) to be assessed	To determine:	Assessment Method	Sample Indicators
Process	Process Management	Monitoring & Reporting	Whether or not a mechanism exists to monitor the iteration progress is monitored	Interviewing	OR2_M30, OR2_D19, OR2_D21, OR2_M31
People	Managers	Buy-in	Whether or not the managers can see that working software is a valid progress indicator	Interviewing	OR2_M32
	Developers	Buy-in	Whether or not the developers can see that working software is a valid progress indicator	Interviewing	OR2_D20

No Big Design up Front (BDUF)

Category of Assessment	Area to be assessed	Characteristic(s) to be assessed	To determine:	Assessment Method	Sample Indicators
Process	Design	Experience	Whether or not design is a continuous process, or done once at the beginning of the development process	Interviewing	OR2_M36, OR2_M37
People	Developers	Buy-in	Whether or not the developers agree to the fact that no big design up front is a valid and efficient approach for agile development	Interviewing	OR2_D22, OR2_D23, OR2_D24
	Managers	Buy-in	Whether managers agree to the fact that no big design up front is a valid and efficient approach for agile development	Interviewing	OR2_M33, OR2_M34, OR2_M35

The Surveys Encompassing the Indicators

Survey for Developers

Indicator	Statements	Nominal Values				
		V	W	X	Y	Z
OR2_D1	Indicate how often are you involved in a project in which all the requirements are not known upfront and an evolutionary requirements approach is used.	Never	Seldom	Sometimes	Usually	Always
OR2_D2	You are willing start a development of a project without knowing the exact requirements of the whole project.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR2_D3	If circumstances dictate that all the details are not available before you start a project, you do not mind the uncertainty and floating targets.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR2_D4	You do not mind starting a project knowing that its requirements will evolve or change in the future.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR2_D5	You can tell the difference between requirements that will the influence the architecture and design of a project and requirements that will not influence it.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR2_D6	In a project, you can recognize high level requirements that most probably will not change versus low level requirements that might change.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR2_D7	Throughout the project the client has full right to change the requirements in order to meet his/her business needs.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR2_D8	In order to deliver valuable software to clients, change should be welcomed but not constrained.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR2_D9	Software development in this organization is not ad hoc or haphazard; there is a clear and known process in place.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR2_D10	Every project involves a clear set of activities. Each of these activities has clear standardized deliverables.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR2_D11	Indicate how often you have worked on a project that was developed in an incremental –iterative approach.	Never	Seldom	Sometimes	Usually	Always
OR2_D12	It is a common practice to divide the system up into mini-projects. The system is seldom developed as one large project.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR2_D13	The incremental-iterative approach has more benefits than the waterfall approach.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree

OR2_D14	You are willing to use the incremental-iterative approach to develop software.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR2_D15	Delivering a working increment every 1-4 weeks will not cause you any additional stress.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR2_D16	No big upfront analysis should be conducted when using the incremental-iterative approach.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR2_D17	You fully understand the principles of the incremental-iterative development approach.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR2_D18	You are willing to do more integration (integrate after each iteration) in order to accommodate the incremental-iterative development approach.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR2_D19	The organization has a usable and efficient method for reporting project status.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR2_D20	Working software should be the primary measure of progress in a project.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR2_D21	During development you deliver a software iteration/release at least once within the organizational status-reporting window (usually one month).	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR2_D22	Development of the first iteration can start without a complete detailed design of the whole system.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR2_D23	Design can start without all the requirements being known except those that are architectural influential.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR2_D24	Design should be revisited before the start of each iteration.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree

Survey for Managers/Executives

Indicator	Statements	Nominal Values				
		V	W	X	Y	Z
OR2_M1	The organization employees know the procedures to gather requirements from clients.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR2_M2	In any project requirements are always gathered from the customer.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR2_M3	Indicate how often you manage a project in which all the requirements are not known upfront and an evolutionary requirements approach is used.	Never	Seldom	Sometimes	Usually	Always
OR2_M4	You can start a development of a project without knowing the exact requirements of the whole project.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR2_M5	If circumstances dictate that all the details are not available before you start a project, you do not mind the uncertainty and floating targets.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR2_M6	You do not mind starting a project knowing that its requirements will evolve or change in the future.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR2_M7	You can tell the difference between requirements that will the influence the architecture and design of a project and requirements that will not influence it.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR2_M8	In a project, you can recognize high level requirements that most probably will not change versus low level requirements that might change.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR2_M9	Throughout the project the client has full right to change the requirements in order to meet his/her business needs.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR2_M10	In order to deliver valuable software to clients change should be welcomed not constrained.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR2_M11	An evolutionary requirements gathering approach could work better than a big upfront approach.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR2_M12	You are willing to try an evolutionary requirements gathering approach.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR2_M13	Software development in this organization is not ad hoc or haphazard; there is a clear and known process in place.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR2_M14	Every project involves a clear set of activities. Each of these activities has clear standardized deliverables.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree

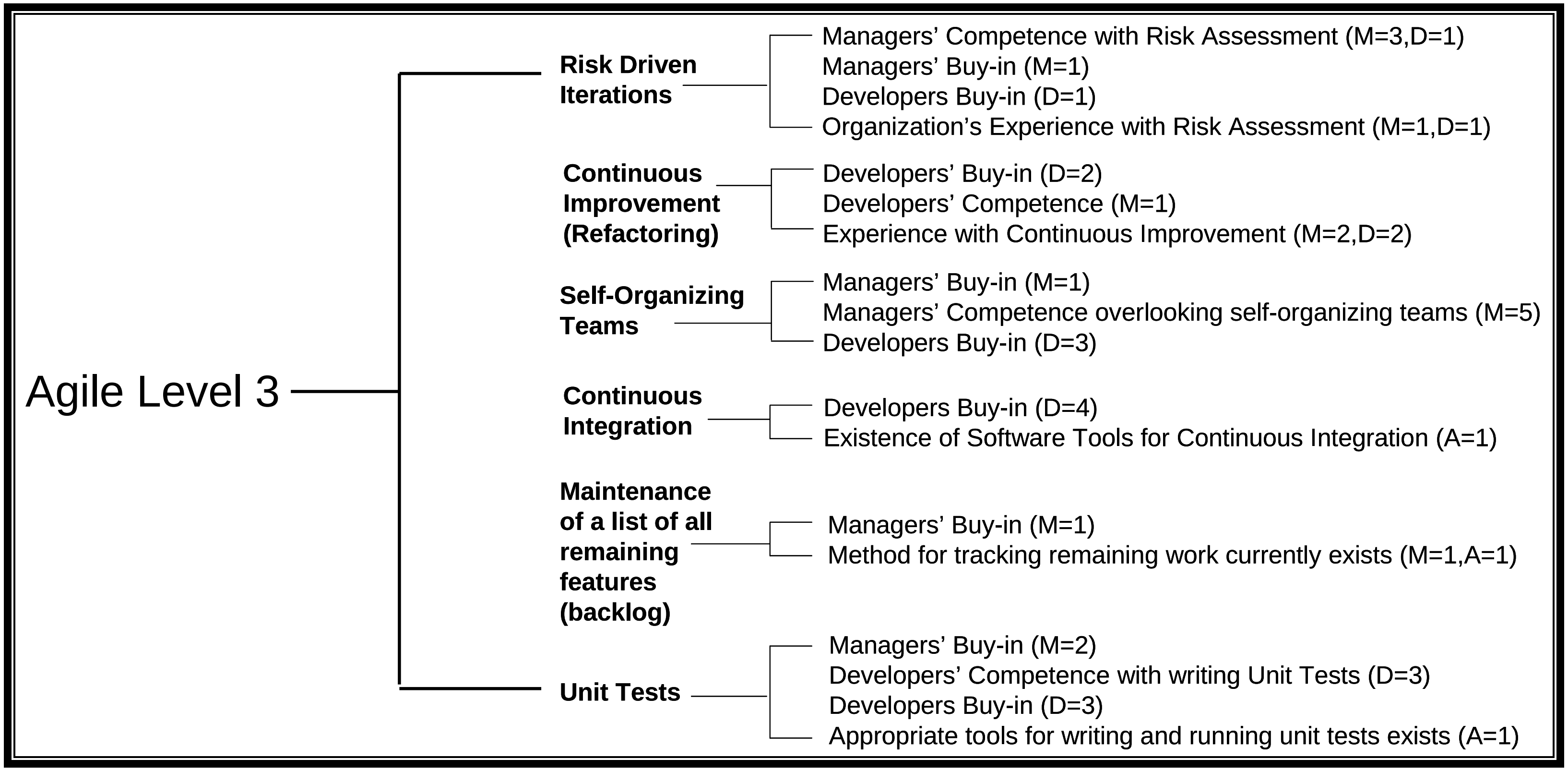
OR2_M15	Indicate how often you develop a project using an incremental–iterative approach.	Never	Seldom	Sometimes	Usually	Always
OR2_M16	It is a common practice to divide the system up into mini-projects. The system is seldom developed as one large project.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR2_M17	The incremental-iterative approach has more benefits than the waterfall approach.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR2_M18	You are willing to use the incremental-iterative approach to develop software.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR2_M19	Delivering a working increment every 1-4 weeks will not cause you any additional stress.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR2_M20	No big upfront analysis should be conducted when using the incremental-iterative approach.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR2_M21	You fully understand the principles of the incremental-iterative development approach.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR2_M22	Planning the project from multiple levels or perspectives (iterations, releases…etc) is better than having one plan for the whole project.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR2_M23	You understand the importance of planning the project in terms of iterations and releases.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR2_M24	You can differentiate between planning features and planning tasks.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR2_M25	Planning for each iteration should occur only right before the actual iteration.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR2_M26	Planning of releases should not be detailed, except for the next/current release.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR2_M27	Indicate your willingness to start a project that is not completely planned out until the end.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR2_M28	Indicate your willingness to commit to planning small iteration and releases continuously through out the project and not to develop one big plan at the beginning of the project.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR2_M29	Indicate how often you create multi-level planning documents when planning a project.	Never	Seldom	Sometimes	Usually	Always
OR2_M30	The organization has a usable and efficient method for reporting project status.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree

OR2_M31	During development you deliver a software iteration/release at least once within the organizational status-reporting window (usually one month).	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR2_M32	Working software should be the primary measure of progress in a project.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR2_M33	Development of the first iteration can start without a complete detailed design of the whole system.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR2_M34	Design can start without all the requirements being know, except those that are architectural influential.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR2_M35	Design should be revisited before the start of each iteration.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR2_M36	In the organization, design is a continuous process that spans the whole development effort and is not done only one time up front.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR2_M37	Indicate how often the organization does not undertake design as a big upfront activity, and instead designs in small increments throughout the development process.	Never	Seldom	Sometimes	Usually	Always

Survey for Assessors

	Statements	Nominal Values				
		V	W	X	Y	Z
OR2_A1	A review of policies and procedures shows that the organization has a process it uses to gather requirements from its clients.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR2_A2	A review of the policies and procedures shows that the organization has a process it uses to develop software. This process should include a set of activities with deliverables and standards.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR2_A3	Inspection of the software development environment shows that the organization has sufficient and useable Software Configuration Tools for agile development.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree

Indicator Map for Agile Level 3


Agile Level 3
Risk Driven Iterations
Managers’ Competence with Risk Assessment (M=3,D=1)
Managers’ Buy-in (M=1)
Developers Buy-in (D=1)
Organization’s Experience with Risk Assessment (M=1,D=1)
Continuous Improvement (Refactoring)
Developers’ Buy-in (D=2)
Developers’ Competence (M=1)
Experience with Continuous Improvement (M=2,D=2)
Self-Organizing Teams
Managers’ Buy-in (M=1)
Managers’ Competence overlooking self-organizing teams (M=5)
Developers Buy-in (D=3)
Continuous Integration
Developers Buy-in (D=4)
Existence of Software Tools for Continuous Integration (A=1)
Maintenance of a list of all remaining features (backlog)
Managers’ Buy-in (M=1)
Method for tracking remaining work currently exists (M=1,A=1)
Unit Tests
Managers’ Buy-in (M=2)
Developers’ Competence with writing Unit Tests (D=3)
Developers Buy-in (D=3)
Appropriate tools for writing and running unit tests exists (A=1)

The number of within the parenthesis indicators denotes the number of indicators used to measure the related organizational characteristic. The letter preceding the number of indicators denotes who should provide the answer to the indicator’s question.

Assessment Tables for Agile Practices within Agile Level 3

Risk Driven Iterations

Category of Assessment	Area to be assessed	Characteristic(s) to be assessed	To determine:	Assessment Method	Sample Indicators
People	Managers	Competence	Whether or not the managers are competent risk assessors	Interviewing	OR3_M1, OR3_M2, OR3_M3, OR3_D2
		Buy-In	Whether or not managers agree to have risks drive the scope of each iteration	Interviewing	OR3_M4
	Developers	Buy-In	Whether or not the developers agree to have risks drive the scope of each iteration	Interviewing	OR3_D3
Process	Risk Assessment	Experience	Whether or not the organization has any experience doing risk assessment or not	Interviewing	OR3_M1, OR3_D1

Continuous Improvement

Category of Assessment	Area to be assessed	Characteristic(s) to be assessed	To determine:	Assessment Method	Sample Indicators
People	Developers	Buy-in	Whether or not the developers agree to adopt an approach of continuous improvement while developing software	Interviewing	OR3_D4, OR3_D5
		Competence	Whether or not the developers are competent enough to refractor code without jeopardizing the existing functionality and quality of the code	Interviewing	OR3_M5
Process	Continuous Improvement	Experience	Whether or not the organization is already involved in continuous improvement	Interviewing	OR3_D6, OR3_D7, OR3_M6, OR3_M7

Continuous Integration

Category of Assessment	Area to be assessed	Characteristic(s) to be assessed	To determine:	Assessment Method	Sample Indicators
People	Developers	Buy-In	Whether or not the developers are willing to commit to continuous integration?	Interviewing	OR3_D14, OR3_D15, OR3_D16, OR3_D17
Environment	Software Tools	Existence	Whether or not the organization has the tools to aid in continuous integration	Observation	OR3_A1

Self Organizing Teams

Category of Assessment	Area to be assessed	Characteristic(s) to be assessed	To determine:	Assessment Method	Sample Indicators
People	Management	Buy-in	Whether or not management agrees to have self-organizing teams	Interviewing	OR3_M11
		Competence	Whether or not management is ready to treat the team as a true self-organizing team	Interviewing	OR3_M8, OR3_M10, OR3_M9, OR3_M12, OR3_M13
	Developers	Buy-In	Whether or not the employees feel comfortable working as self-organizing teams	Interviewing	OR3_D8, OR3_D9, OR3_D10

Maintenance of a List of All Remaining Features (Backlog)

Category of Assessment	Area to be assessed	Characteristic(s) to be assessed	To determine:	Assessment Method	Sample Indicators
People	Management	Buy-in	Whether or not management is willing to maintain an up-to-date list of all the remaining features for the project (backlog)	Interviewing	OR3_M16
Process	Project Management	Existence	Whether or not the organization currently keeps an up-to-date list of all the work that remains to be done	Interviewing	OR3_M15
				Observation	OR3_A2

Unit Tests

Category of Assessment	Area to be assessed	Characteristic(s) to be assessed	To determine:	Assessment Method	Sample Indicators
People	Developers	Buy-in	Whether or not developers are willing to write unit tests during the development process	Interviewing	OR3_D18, OR3_D19, OR3_D21
		Competence	Whether or not the developers have the competence and previous experience writing unit tests	Interviewing	OR3_D20, OR3_M19, OR3_D22
	Managers	Buy-In	Whether or not the management accepts that developers will invest additional time to write unit tests while coding	Interviewing	OR3_M17, OR3_M18
Environment	Software Tools	Existence	Whether or not the organization has the tools that support writing and running unit tests	Observation	OR3_A3

The Surveys Encompassing the Indicators

Survey for Developers

Indicator	Statements	Nominal Values				
		V	W	X	Y	Z
OR3_D1	For the projects that you have worked on, indicate how often risk assessment was performed during the project and communicated to the whole team.	Never	Seldom	Sometimes	Usually	Always
OR3_D2	Your manager is very competent when coming to risk assessments and mitigation plans.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR3_D3	The riskiest, most difficult elements should be approached first in the early iterations of the development effort.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR3_D4	It is important to put effort into improving the design and code of a component, even if it is already working.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR3_D5	You are willing to adopt an approach of continuous improvement during software development.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR3_D6	It is a common practice in the organization to revisit a working component to improve its design or code structure.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR3_D7	Indicate how often you revisit a working component to improve its design or code structure.	Never	Seldom	Sometimes	Usually	Always
OR3_D8	You like to work on a team that management regards as one entity; not addressing individual team members in rewards or tasks, but as one team.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR3_D9	You do not mind working without direct managerial supervision as long as you are on a team that is treated as a partner with management.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR3_D10	You consider yourself competent and disciplined enough to work on self-organizing teams	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR3_D11	Indicate how often you develop software projects using the Object Oriented (OO) principles and techniques.	Never	Seldom	Sometimes	Usually	Always
OR3_D12	You understand the OO principles and theories very well.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree

OR3_D13	Indicate how often the organization takes the OO approach in development of software projects.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR3_D14	The usual time it takes to create a build the system is:	More than 1 hour	Under 1 hour	Under 15 minutes	Under 10 minutes	Under 5 minutes
OR3_D15	Instead of integrating the system at the end of the development effort, it is better to regularly integrate the system throughout the whole development process.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR3_D16	You are trained to use the Software Configuration Management tool for continuous integration.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR3_D17	You are willing to integrate your software throughout the development process, even if it means more work for you.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR3_D18	It is important to write unit tests for methods and functions while coding them even if that will take additional time.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR3_D19	Writing unit tests for code is as important as writing new code for more functionality.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR3_D20	Indicate how often you write unit tests for every method or function in your code.	Never	Seldom	Sometimes	Usually	Always
OR3_D21	You are willing to commit to writing unit tests while you code for every method or function in your code.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR3_D22	You consider yourself competent enough to write good and comprehensive unit tests for the methods and functions in your code.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree

Survey for Managers/Executives

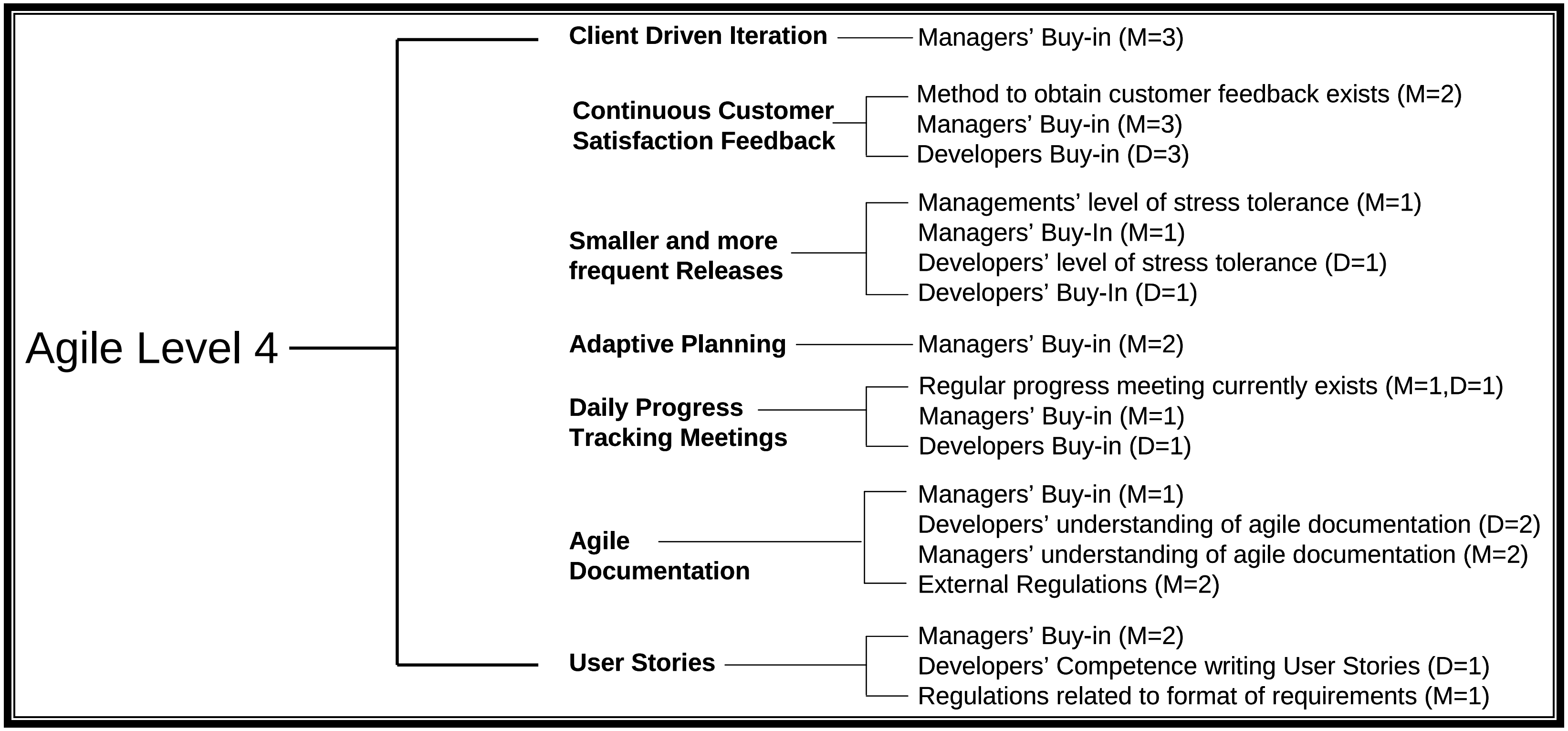
Indicator	Statements	Nominal Values				
		V	W	X	Y	Z
OR3_M1	Indicate how often do you perform risk assessment and mitigation techniques during a project.	Never	Seldom	Sometimes	Usually	Always
OR3_M2	You have been trained to perform risk assessments.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR3_M3	You are very competent performing risk assessment and mitigation plans.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR3_M4	The riskiest, most difficult elements should be approached first in the early iterations of the development effort.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR3_M5	The developers are competent enough to refractor code without jeopardizing the existing functionality and quality and breaking any unit tests (if they exist).	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR3_M6	It is a common practice in the organization to revisit a working component to improve its design or code structure.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR3_M7	Indicate how often you make sure that your subordinates revisit a working component to improve its design or code structure.	Never	Seldom	Sometimes	Usually	Always
OR3_M8	You can trust your employees' capabilities to determine the best way to accomplish tasks by themselves without your (management‘s) interference.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR3_M9	Employees are competent and disciplined enough to work in self-organizing teams.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR3_M10	You are willing to allow space for the self-organizing team to grow and not micromanage it.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR3_M11	You agree that it is very important for the employees to work in teams where they can divide the team tasks among themselves.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR3_M12	The team is an entity that has its knowledge, perspective, motivation and expertise and should be treated as a partner with management and the customer.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR3_M13	The self-organizing team can negotiate commitments.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR3_M14	Indicate how often your organization takes the OO approach in software development	Never	Seldom	Sometimes	Usually	Always

OR3_M15	When working on a project you keep an up-to-date list of all the work that remains to be done.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR3_M16	You are willing to keep an up-to-date list of all the work that remains to be done.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR3_M17	It is important for developers to write unit tests for their methods and functions while they code, even if that will take additional time from them.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR3_M18	Writing unit tests for code is as important as writing new code for more functionality.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR3_M19	The developers are competent enough to write good unit tests for the methods and functions in the code.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree

Survey for Assessors

	Statements	Nominal Values				
		V	W	X	Y	Z
OR3_A1	After looking at the software development tools, you know that the organization has the SCM tools and processes to support continuous integration.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR3_A2	After inspecting previous projects, you know that each project had a mechanism by which all the remaining work in a project was known at any point in time.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR3_A3	After looking at the software development tools, you know that the organization has the necessary tools to write and run unit tests within the development IDE.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree

Indicator Map for Agile Level 4


Agile Level 4
Client Driven Iteration
Managers’ Buy-in (M=3)
Continuous Customer Satisfaction Feedback
Method to obtain customer feedback exists (M=2)
Managers’ Buy-in (M=3)
Developers Buy-in (D=3)
Smaller and more frequent Releases
Managements’ level of stress tolerance (M=1)
Managers’ Buy-In (M=1)
Developers’ level of stress tolerance (D=1)
Developers’ Buy-In (D=1)
Adaptive Planning
Managers’ Buy-in (M=2)
Daily Progress Tracking Meetings
Regular progress meeting currently exists (M=1,D=1)
Managers’ Buy-in (M=1)
Developers Buy-in (D=1)
Agile Documentation
Managers’ Buy-in (M=1)
Developers’ understanding of agile documentation (D=2)
Managers’ understanding of agile documentation (M=2)
External Regulations (M=2)
User Stories
Managers’ Buy-in (M=2)
Developers’ Competence writing User Stories (D=1)
Regulations related to format of requirements (M=1)

The number of within the parenthesis indicators denotes the number of indicators used to measure the related organizational characteristic. The letter preceding the number of indicators denotes who should provide the answer to the indicator’s question.

Assessment Tables for Agile Practices within Agile Level 4

Client Driven Iterations

Category of Assessment	Area to be assessed	Characteristic(s) to be assessed	To determine:	Assessment Method	Sample Indicators
People	Managers	Buy-in	Whether or not managers are willing to give the customer the power to dictate the scope of the iterations	Interviewing	OR4_M1, OR4_M2, OR4_M3

Continuous Customer Satisfaction Feedback

Category of Assessment	Area to be assessed	Characteristic(s) to be assessed	To determine:	Assessment Method	Sample Indicators
Process	Customer Feedback	Existence	Whether or not the organization has a method by which they gather continuous feedback/criticism from the customer during the development process	Interviewing	OR4_M4, OR4_M5
People	Developers	Buy-in	Whether or not the developers accept the fact that the customers are encouraged to continually re-think their requirements	Interviewing	OR4_D1, OR4_D2, OR4_D3
	Managers	Buy-in	Whether or not the managers accept the fact that the customers are encouraged to continually re-think their requirements	Interviewing	OR4_M2, OR4_M6, OR4_M7

Smaller and more Frequent Releases

Category of Assessment	Area to be assessed	Characteristic(s) to be assessed	To determine:	Assessment Method	Sample Indicators
People	Managers	Buy-in	Whether or not the managers understand the importance of having smaller and more frequent releases to give the customer quicker feedback	Interviewing	OR4_M12
		Stress	Whether or not managers can handle the additional stress of overseeing the delivery of fully functional releases every 4-8 weeks	Interviewing	OR4_M13
	Developers	Buy-in	Whether or not the developers understand the importance of having smaller and more frequent releases to give the customer quicker feedback	Interviewing	OR4_D8
		Stress	Whether or not the developers can handle the increased stress of delivering fully functional releases every 4-8 weeks	Interviewing	OR4_D9

Adaptive Planning

Category of Assessment	Area to be assessed	Characteristic(s) to be assessed	To determine:	Assessment Method	Sample Indicators
People	Management	Buy-in	Whether or not management is willing to base the planning for the next iteration on the client's feedback from the current (previous) iteration	Interviewing	OR4_M14
			Whether or not management is willing to plan as late as possible for an iteration (immediately before the iteration)	Interviewing	OR4_M15

Daily Progress Tracking Meetings

Category of Assessment	Area to be assessed	Characteristic(s) to be assessed	To determine:	Assessment Method	Sample Indicators
People	Management	Buy-In	Whether or not management is willing to meet daily for progress update	Interviewing	OR4_M16
	Developers	Buy-In	Whether or not the developers are willing to meet daily for progress updates	Interviewing	OR4_D10

Process	Project management	Progress meetings	How often the team meets regularly to discuss the progress of a project	Interviewing	OR4_M17, OR4_D11

Agile Documentation (from Agile Modeling)

Category of Assessment	Area to be assessed	Characteristic(s) to be assessed	To determine:	Assessment Method	Sample Indicators
People	Developers	Competence	Whether or not the developers understand what an Agile approach to documentation is	Interviewing	OR4_D12, OR4_D13
	Management	Competence	Whether or not management understands what an Agile approach to documentation is	Interviewing	OR4_M18, OR4_M19
		Buy-In	Whether or not management is willing to take an Agile approach to documentation	Interviewing	OR4_M20
Process	Documentation	Regulations	Whether or not external regulatory requirements exist that dictate the production of heavy (detailed) documentation for every aspect of the process	Interviewing	OR4_M21, OR4_M22

User Stories

Category of Assessment	Area to be assessed	Characteristic(s) to be assessed	To determine:	Assessment Method	Sample Indicators
People	Management	Buy-In	Whether or not management is willing to use user stories as an elicitation method/form for high level requirements	Interviewing	OR4_M23, OR4_M24
	Developers	Competence	Whether or not the developers have the understanding/knowledge of how to use user stories	Interviewing	OR4_D14
Process	Requirements	Regulations	Whether or not there are regulatory requirements for the elicitation of the requirements (they have to specified in a certain form)	Interviewing	OR4_M25

The Surveys Encompassing the Indicators

Survey for Developers

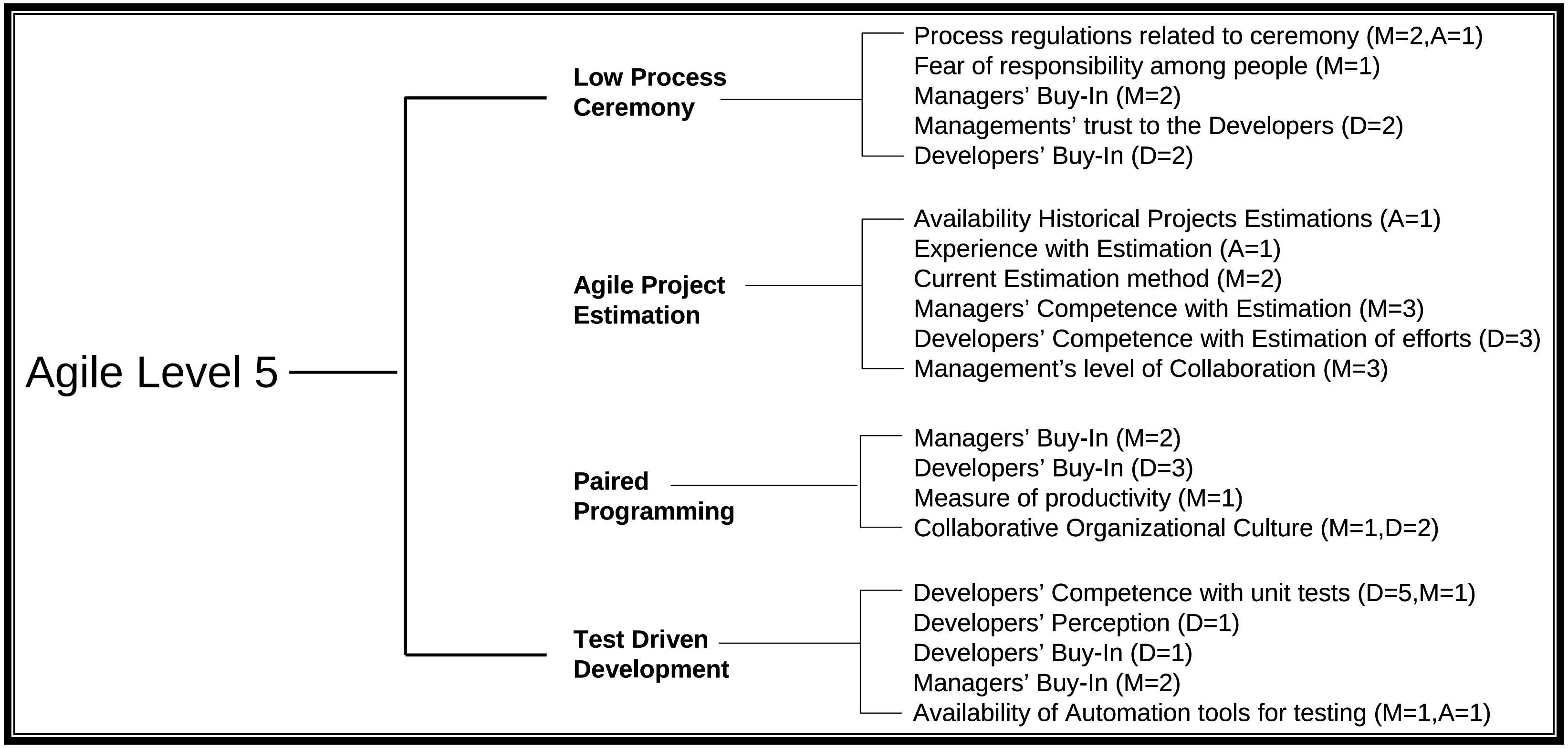
Indicator	Statements	Nominal Values				
		V	W	X	Y	Z
OR4_D1	Customers should be encouraged to regularly change their expectations for the product being developed to ensure that the product satisfies the business priorities of the organization.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR4_D2	As the perception of what they need changes, customers are expected to articulate these changes and thus affect the product being built.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR4_D3	The customer should give his/her feedback throughout the development process even if it means that requirements must be changed.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR4_D8	Smaller and more frequent releases are important in order to give the customer a means by which he/she can give more and quicker feedback.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR4_D9	Delivering smaller and more frequent fully functional releases every 4-8 weeks will not cause you any additional stress.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR4_D10	You are willing to meet daily to check in and synchronize efforts with your team members.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR4_D11	Indicate how often you meet with the rest of the team to discuss and update each other about the progress of the project.	Less frequent than monthly	Monthly	Every couple of weeks	Weekly	Daily/Hourly
OR4_D12	Documentation exists within an Agile development process	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR4_D13	You understand the role of documentation within an Agile development process	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR4_D14	You can use user stories instead of requirements to develop the architectural framework of the system.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree

Survey for Managers/Executives

Indicator	Statements	Nominal Values				
		V	W	X	Y	Z
OR4_M1	As the perception of what they need changes, customers are expected to articulate those changes by prioritizing the features they would like to see in the next iteration.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR4_M2	Customers should be encouraged to regularly change their expectations for the product being developed to ensure that the product satisfies the business priorities of the organization.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR4_M3	The customer should be given the authority to direct what is being developed in which iteration.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR4_M4	The customer has the opportunity to give his/her feedback about the product through out the development process by means of interacting with a working piece of software or a least a prototype.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR4_M5	The organization has a method by which it gathers continuous feedback/criticism from the customer during the development process.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR4_M6	As the perception of what they need changes, customers are expected to articulate those changes and so affect the product being built.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR4_M7	The customer should give his/her feedback throughout the development process even if it means that requirements must be changed.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR4_M12	Smaller and more frequent releases are important in order to give the customer a means by which he/she can give more and quicker feedback.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR4_M13	Delivering smaller and more frequent fully functional releases every 4-8 weeks will not cause you any additional stress.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR4_M14	The plan for upcoming iteration may change based on customer feedback from the previous or current iteration.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR4_M15	You agree with developing the detailed plan for an iteration only after the conclusion of the previous iteration.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR4_M16	You are willing to meet daily for the progress update of a project.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR4_M17	Indicate how often you meet with the rest of the team to discuss and update each other on the progress of the project.	Less frequent than monthly	Monthly	Every couple of weeks	Weekly	Daily/Hourly
OR4_M18	Documentation exists within an Agile development process.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree

OR4_M19	You understand the role of documentation within an Agile development process.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR4_M20	You will allow your subordinates to take an Agile approach to documentation.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR4_M21	Stakeholders do not require heavy (detailed) documentation for any activities or aspects of the development process.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR4_M22	You are not required by any external auditory to maintain fine heavy (detailed) documentation for activities or aspects of the development process.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR4_M23	You are willing to adopt user stories as a method for high level requirements elicitation.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR4_M24	User stories can be used instead of large requirements documents.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR4_M25	No regulatory constraints exist that prevent the use of user stories as a means of capturing high level requirements from the user.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree

Indicator Map for Agile Level 5



Agile Level 5

- **Low Process Ceremony**
  - Process regulations related to ceremony (M=2,A=1)
  - Fear of responsibility among people (M=1)
  - Managers’ Buy-In (M=2)
  - Managements’ trust to the Developers (D=2)
  - Developers’ Buy-In (D=2)
- **Agile Project Estimation**
  - Availability Historical Projects Estimations (A=1)
  - Experience with Estimation (A=1)
  - Current Estimation method (M=2)
  - Managers’ Competence with Estimation (M=3)
  - Developers’ Competence with Estimation of efforts (D=3)
  - Management’s level of Collaboration (M=3)
- **Paired Programming**
  - Managers’ Buy-In (M=2)
  - Developers’ Buy-In (D=3)
  - Measure of productivity (M=1)
  - Collaborative Organizational Culture (M=1,D=2)
- **Test Driven Development**
  - Developers’ Competence with unit tests (D=5,M=1)
  - Developers’ Perception (D=1)
  - Developers’ Buy-In (D=1)
  - Managers’ Buy-In (M=2)
  - Availability of Automation tools for testing (M=1,A=1)

The number of within the parenthesis indicators denotes the number of indicators used to measure the related organizational characteristic. The letter preceding the number of indicators denotes who should provide the answer to the indicator’s question.

Assessment Tables for Agile Practices within Agile Level 5

Low Process Ceremony (Process Ceremony is the level of paperwork involved with a process)

Category of Assessment	Area to be assessed	Characteristic(s) to be assessed	To determine:	Assessment Method	Sample Indicators
Process	Ceremony	Regulations	Whether or not the organization needs to maintain a high process ceremony due to certain audits or regulations	Interviewing	OR5_M1, OR5_M2
				Observation	OR5_A1
Culture	Organizational	Responsibility	Whether or not there is a fear of responsibility/blame among people, thus supporting the high level of process ceremony	Interviewing	OR5_M3
People	Developers	Buy-in	Whether or not the developers feel comfortable decreasing the level of process ceremony	Interviewing	OR5_D1, OR5_D2
	Management	Buy-in	Whether or not the managers feel comfortable decreasing the level of process ceremony	Interviewing	OR5_M4, OR5_M5
		Trust	Whether or not the management trusts the developers to make decisions on their own without their “approval”	Interviewing	OR5_M6, OR5_M7

Agile Project Estimation

Category of Assessment	Area to be assessed	Characteristic(s) to be assessed	To determine:	Assessment Method	Sample Indicators
Process	Estimation	Experience	Whether or not the organization is experienced in estimation	Observation	OR5_A2
		Existence	Whether or not data exists from previous projects to aid with the estimation	Observation	OR5_A3
		Method	Whether or not the estimation process separates the estimation of effort from the estimation of duration	Interviewing	OR5_M8, OR5_M9
People	Developers	Competence	Whether or not the developers are competent in making their own estimates of effort.	Interviewing	OR5_D3, OR5_D4, OR5_D5
	Management	Competence	Whether or not the managers are competent in making estimates.	Interviewing	OR5_M10, OR5_M11, OR5_M12
	Management	Collaboration	Whether management will encourage the estimation process to be done by the whole team or by only them	Interviewing	OR5_M13, OR5_M14, OR5_M15

Paired Programming

Category of Assessment	Area to be assessed	Characteristic(s) to be assessed	To determine:	Assessment Method	Sample Indicators
People	Management	Buy-in	Whether or not management can see the benefit from paired programming	Interviewing	OR5_M16, OR5_M17
	Developers	Buy-in	Whether or not developers are willing to try paired programming	Interviewing	OR5_D6, OR5_D7, OR5_D8
Process	Project Management	Measurement of Productivity	What the organization considers to be a measure of software productivity	Interviewing	OR5_M18
Culture	Organizational	Collaboration	Whether or not an atmosphere of assistance exists in the organization	Interviewing	OR5_D9, OR5_D10, OR5_M19

Test Driven Development

Category of Assessment	Area to be assessed	Characteristic(s) to be assessed	To determine:	Assessment Method	Sample Indicators
People	Developers	Competence	Whether or not the developers are competent and experienced with writing unit tests	Interviewing	OR5_D11, OR5_D12, OR5_D13
			Whether or not the developers have a very strong understanding of OO concepts	Interviewing	OR5_D14, OR5_D15, OR5_M20
		Buy-In	Whether or not the developers are motivated and willing to apply test driven development	Interviewing	OR5_D16
		Perception	Whether or not the developers think that Test-driven development is a hard task or not	Interviewing	OR5_D17
	Management	Buy-In	Whether or not management will encourage test-driven development and tolerate the learning curve	Interviewing	OR5_M21. OR5_M22
Environment	Software Tools	Test Automation	Whether or not the organization has or can provide tools for creating and maintaining automated test suites	Observation	OR5_A4
				Interviewing	OR5_M23

The Surveys Encompassing the Indicators

Survey for Developers

Indicator	Statements	Nominal Values				
		V	W	X	Y	Z
OR5_D1	You favor accepting responsibility and being held accountable when things go wrong over multiple layers of formal steps, reviews, and procedures.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR5_D2	You do not support the existence of various formal steps and reviews to reduce (spread) accountability when something goes wrong.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR5_D3	Indicate how often you make size/effort estimates for the project or a component of the project that you will be working on.	Never	Seldom	Sometimes	Usually	Always
OR5_D4	You have been trained on how to make feature estimates.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR5_D5	You are competent enough to make your own estimates of size/effort.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR5_D6	Paired programming increases productivity contrary to what others say about paired programming decreasing productivity by half.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR5_D7	Indicate how often you program in pairs.	Never	Seldom	Sometimes	Usually	Always
OR5_D8	You are willing to program in pairs.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR5_D9	An atmosphere of assistance exists in the organization.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR5_D10	Whenever you need help people are willing to help you.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR5_D11	Indicate how often you write unit tests for every function or method one writing code.	Never	Seldom	Sometimes	Usually	Always
OR5_D12	You have no problems or challenges writing unit tests for functions and methods.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree

OR5_D13	The suite of unit tests that you write is comprehensive and usually encompasses all possible test scenarios.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR5_D14	Indicate how often you program in the object-oriented (OO) paradigm.	Never	Seldom	Sometimes	Usually	Always
OR5_D15	You have a very strong understanding of object-oriented concepts and principles.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR5_D16	You are willing to employ a test-driven approach to development.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR5_D17	Test-driven development is easy.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree

Survey for Assessors

Indicator	Statements	Nominal Values				
		V	W	X	Y	Z
OR5_A1	A review of policies and procedures shows that there is no need for a high process ceremony in this organization.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR5_A2	A review of previous project documentation shows that the effort estimates were within acceptable range to the actual effort that was put into delivering the project.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR5_A3	Previous project documentation, including effort and size estimations, are available and accessible.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR5_A4	An inspection of these software tools shows that the organization has accessible and usable tools for creating and maintaining automated test suites.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree

Survey for Managers/Executives

Indicator	Statements	Nominal Values				
		V	W	X	Y	Z
OR5_M1	There are no regulations or auditory requirements that dictate the need for high process ceremony.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR5_M2	Your organization is informal and flexible. There are not many formal steps, policies or procedures.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR5_M3	People in the organization are not afraid of taking responsibility.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR5_M4	You favor accepting responsibility and being held accountable when things go wrong over multiple layers of formal steps, reviews, and procedures.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR5_M5	You do not support the existence of various formal steps and reviews to reduce (spread) accountability when something goes wrong.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR5_M6	You trust your subordinates to make decisions within their scope of work without referring back to you for approval.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR5_M7	Your subordinates are competent to make their own decisions without referring back to you for approval.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR5_M8	When preparing a project estimation, you estimate the size first and derive from that a duration estimate.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR5_M9	The estimation process employed by the organization separates the estimation of effort from the estimation of duration.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR5_M10	Indicate how often you make size/effort estimates for projects.	Never	Seldom	Sometimes	Usually	Always
OR5_M11	You have been trained on how to make project estimates.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR5_M12	You are competent and experienced enough to make realistic estimates of size/effort.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR5_M13	The whole team participating in project estimation will yield better and more accurate results.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR5_M14	Indicate how often the whole team has participated in creating project estimates.	Never	Seldom	Sometimes	Usually	Always
OR5_M15	You will encourage the whole development team to actively participate in developing a project estimate.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree

OR5_M16	Paired programming increases productivity contrary to what others say about paired programming decreasing productivity by half.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR5_M17	You encourage your development team to use paired programming.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR5_M18	Productivity is about how much customer value can you create per dollar spent not about how many lines of code, classes coded or Use Cases per dollar spent.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR5_M19	An atmosphere of assistance exists in the organization.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR5_M20	The development team has a very strong understanding of object-oriented concepts and principles.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR5_M21	Test-driven development will produce better software with fewer bugs	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR5_M22	You are willing to tolerate the learning curve of the development team while they transition to test-driven development.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree
OR5_M23	The organization will be willing to provide software tools for creating and maintaining automated test suites.	Strongly Disagree	Tend to Disagree	Neither Agree nor Disagree	Tend to Agree	Strongly Agree

Appendix B: Empty and Completed Surveys

This appendix includes the survey’s used during the substantiation process and the results obtained. First an empty 2-page survey, used to gather high-level feedback about the measurement index and the 4-stage process, is presented. It is proceeded by 28 completed surveys. Then an empty 12-page survey which was used for gathering detailed feedback is presented along with 2 ones that were completed.

Empty 2-Page Survey (Overview)

VIRGINIA POLYTECHNIC INSTITUTE AND STATE UNIVERSITY

Assessment Questionnaire:
“Process for Adoption of Agile Practices in Projects”

Date:	Reference # *(for archiving purposes):*

SECTION 1: ASSESSOR’S INFORMATION

Name (Optional):

Organization / Institute:

Official Position:	**Years in position :**
Email :	**Phone Number :**

Please rate your familiarity with Agile software development	**1** Not familiar	**2**	**3**	**4** somewhat familiar	**5**	**6**	**7** expert
Please rate your highest level of involvement in development efforts that used Agile practices	**1** None	**2**	**3**	**4** participant	**5**	**6**	**7** leader
How frequently have you aided entities in adopting Agile practices	**1** Never	**2**	**3**	**4** occasionally	**5**	**6**	**7** constantly
Please rate your level of familiarity with general process assessment and/or process improvement	**1** None	**2**	**3**	**4** participant	**5**	**6**	**7** leader

SECTION 2: AGILE LEVELS

	strongly disagree	slightly disagree	neither disagree nor agree	slightly agree	strongly agree
COMPREHENSIVENESS					
The 5 Agile Levels are comprehensive enough to depict the various states and levels most organizations could be at relative to agility?					
The 5 Agile Levels are defined in a valid and logical order?					
PRACTICALITY					
One of our objectives is to make sure that the agile levels are practical and can be used in industry. In light of this, to what extent would you agree that these Agile Levels can be used to classify, and/or aid in the transition of, currently employed software development efforts?					
NECESSITY					
The classification of agility into five Agile Levels as presented would be beneficial to the software engineering industry?					
RELEVANCE					
Each of the Agile Levels presented encompass a set of agile practices and concepts. To what extent would you agree that those practices and concepts are relevant and correctly assigned to their respective agile levels?					

Would you add, remove or redefine any of the 5 agile levels? If so please explain why.

Do you of any further comments about the Agile Levels?

SECTION 3: THE OVERALL PROCESS FRAMEWORK

	strongly disagree	slightly disagree	neither disagree nor agree	slightly agree	strongly agree
UNDERSTANDABILITY					
For the topics listed below designate the degree to which you agree that they are understandable					
Overall objective of this process framework					
Discontinuing Factors					
5 Agile Levels					
Project –Level Assessment					
Organizational Assessment					
Gap Analysis					
PRACTICALITY					
One of our objectives is to make sure that the process framework is practical and can be used in industry. In light of this to what extent would you agree that process framework would be practical and feasible to employ?					
NECESSITY					
The process framework is beneficial to the software engineering industry?					
COMPLETENESS					
All the necessary components are present in this process framework in order to achieve its overall goal of aiding an organization in the adoption of agile development for its various projects?					
The steps and activities in the process framework are organized in a logical and valid sequence in order to achieve its overall goal of an assisting agile adoption?					
EFFECTIVENESS					
To what extent do you agree that each of the components in this process framework is necessary and sufficient for the framework to achieve its purpose?					
Discontinuing Factors					
5 Agile Levels					
Project –Level Assessment					
Organizational Assessment					
Gap Analysis					

Would you add, remove or redefine any components of this process framework? If so please explain why.

____

____

____

____

Do you have any additional comments about the process framework?

____

____

____

____

SECTION 5: FURTHER EVALUTION

Would you like to participate in an extensive, more detailed, evaluation of this research?	**YES**	**NO**	**MAYBE**

Completed 2-page Surveys (Overview)

VIRGINIA POLYTECHNIC INSTITUTE AND STATE UNIVERSITY

Virginia Tech

Assessment Questionnaire:
"Process for Adoption of Agile Practices in Projects"

OV01

Date:	Reference # *(for archiving purposes)*:

SECTION 1: ASSESSOR'S INFORMATION

Name (Optional): [redacted]

Organization / Institute: [redacted] → Agile Coach, Instructor

Official Position: Principal (Consultant) **Years in position :** 8 years

Email : [redacted] **Phone Number :**

Please rate your familiarity with Agile software development	1 Not familiar 2 3 4 somewhat familiar 5 6 [✓ 7] expert
Please rate your highest level of involvement in development efforts that used Agile practices	1 None 2 3 4 participant 5 6 [✓ 7] leader
How frequently have you aided entities in adopting Agile practices	1 Never 2 3 4 occasionally 5 6 [✓ 7] constantly
Please rate your level of familiarity with general process assessment and/or process improvement	1 None 2 3 4 participant 5 6 [✓ 7] leader

SECTION 2: AGILE LEVELS

The 5 Agile Levels are comprehensive enough to depict the various states and levels most organizations could be at relative to agility?					○
The 5 Agile Levels are defined in a valid and logical order?					○
One of our objectives is to make sure that the agile levels are practical and can be used in industry. In light of this, to what extent would you agree that these Agile Levels can be used to classify, and/or aid in the transition of, currently employed software development efforts?					○
The classification of agility into five Agile Levels as presented would be beneficial to the software engineering industry?					○
Each of the Agile Levels presented encompass a set of agile practices and concepts. To what extent would you agree that those practices and concepts are relevant and correctly assigned to their respective agile levels?				✓	

Would you add, remove or redefine any of the 5 agile levels? If so please explain why.

1) Merge Iteration & Communication into feedback. This is in answering the question 'What one thing would you introduce?'

2) Add a smell-driven adoption approach. That means to adopt practices based on customer value.

Do you of any further comments about the Agile Levels?

~~They are a [illegible]~~ They have a "rightness" about them. They fit with my experience to a point. They are however not customer focused. This will, in my opinion, affect the effectiveness of their introduction

Organizational level would be better as a 'project perimeter'

SECTION 3: THE OVERALL PROCESS FRAMEWORK

Ovel

	Strongly disagree	Slightly disagree	neither disagree nor agree	Slightly agree	Strongly agree
[illegible]					
Overall objective of this process framework					✓
Discontinuing Factors				○	
5 Agile Levels					○
Project –Level Assessment					✓
Organizational Assessment				✓	
Gap Analysis					○
[illegible]					
One of our objectives is to make sure that the process framework is practical and can be used in industry. In light of this to what extent would you agree that process framework would be practical and feasible to employ?				○	
[illegible]					
The process framework is beneficial to the software engineering industry?				✓	
[illegible]					
All the necessary components are present in this process framework in order to achieve its overall goal of aiding an organization in the adoption of agile development for its various projects?				✓	
The steps and activities in the process framework are organized in a logical and valid sequence in order to achieve its overall goal of an assisting agile adoption?				✓	
[illegible]					
Discontinuing Factors		✓			
5 Agile Levels				✓	
Project –Level Assessment					✓
Organizational Assessment				✓	
Gap Analysis					○

Would you add, remove or redefine any components of this process framework? If so please explain why.

1) Weaken the Discontinuing factors → always a way to add value

2) Modify ORG assesment to perimeter assessment.

Do you have any additional comments about the process framework?

This is the RIGHT time for this work! Excellent job.

SECTION 5: FURTHER EVALUTION

[illegible] Research?	[✓ YES]	NO	MAYBE

OVØ2

VIRGINIA POLYTECHNIC INSTITUTE AND STATE UNIVERSITY

Virginia Tech

Assessment Questionnaire:
"Process for Adoption of Agile Practices in Projects"

Date:	Reference # *(for archiving purposes)*:

SECTION 1: ASSESSOR'S INFORMATION

Name (Optional): [redacted]

Organization / Institute:

Official Position: Director | **Years in position :** 2 years

Email : [redacted] | **Phone Number :** [redacted]

Please rate your familiarity with Agile software development	1 Not familiar	2	3	4 somewhat familiar	5	6	[✓ 7] expert
Please rate your highest level of involvement in development efforts that used Agile practices	1 None	2	3	4 participant	5	6	[✓ 7] leader
How frequently have you aided entities in adopting Agile practices	1 Never	2	3	4 occasionally	5	6	[✓ 7] constantly
Please rate your level of familiarity with general process assessment and/or process improvement	1 None	2	3	4 participant	5	[✓ 6]	7 leader

SECTION 2: AGILE LEVELS

The 5 Agile Levels are comprehensive enough to depict the various states and levels most organizations could be at relative to agility? projects					✓
The 5 Agile Levels are defined in a valid and logical order?				✓	
One of our objectives is to make sure that the agile levels are practical and can be used in industry. In light of this, to what extent would you agree that these Agile Levels can be used to classify, and/or aid in the transition of, currently employed software development efforts?				✓	
The classification of agility into five Agile Levels as presented would be beneficial to the software engineering industry?				✓	
Each of the Agile Levels presented encompass a set of agile practices and concepts. To what extent would you agree that those practices and concepts are relevant and correctly assigned to their respective agile levels?				✓	

Would you add, remove or redefine any of the 5 agile levels? If so please explain why.

I think that they are OK at the project level

Do you of any further comments about the Agile Levels?

OV02

SECTION 3: THE OVERALL PROCESS FRAMEWORK

	Strongly disagree	Slightly disagree	Neither disagree nor agree	Slightly agree	Strongly agree
[illegible]					
Overall objective of this process framework				✓	
Discontinuing Factors				✓	
5 Agile Levels				✓	
Project –Level Assessment				✓	
Organizational Assessment			✓		
Gap Analysis				✓	
[illegible]					
One of our objectives is to make sure that the process framework is practical and can be used in industry. In light of this to what extent would you agree that process framework would be practical and feasible to employ?				✓	
[illegible]					
The process framework is beneficial to the software engineering industry?				✓	
[illegible]					
All the necessary components are present in this process framework in order to achieve its overall goal of aiding an organization in the adoption of agile development for its various projects?		✓			
The steps and activities in the process framework are organized in a logical and valid sequence in order to achieve its overall goal of an assisting agile adoption?				✓	
[illegible]					
Discontinuing Factors				✓	
5 Agile Levels				✓	
Project –Level Assessment				✓	
Organizational Assessment				✓	
Gap Analysis				✓	

Would you add, remove or redefine any components of this process framework? If so please explain why.

organizational assessment could use more work. –
examples: internal & external auditors, HR, internal metrics, etc
+

Do you have any additional comments about the process framework?

SECTION 5: FURTHER EVALUTION

[illegible] research?	YES	NO	[✓ MAYBE]

Note :- Business Value
Diss.

VIRGINIA POLYTECHNIC INSTITUTE AND STATE UNIVERSITY

Virginia Tech

Assessment Questionnaire:
"Process for Adoption of Agile Practices in Projects"

OV03

Date: 10/13/06	Reference # *(for archiving purposes)*:

SECTION 1: ASSESSOR'S INFORMATION

Name (Optional):

Organization / Institute:

Official Position: DIRECTOR, LEAN-AGILE PRACTICE **Years in position :** 5

Email : **Phone Number :**

Please rate your familiarity with Agile software development	1 Not familiar	2	3	4 somewhat familiar	5	6	[✓ 7] Expert
Please rate your highest level of involvement in development efforts that used Agile practices	1 None	2	3	4 participant	5	6	[✓ 7] leader
How frequently have you aided entities in adopting Agile practices	1 Never	2	3	4 occasionally	5	6	[✓ 7] constantly
Please rate your level of familiarity with general process assessment and/or process improvement	1 None	2	3	4 participant	5	6	[✓ 7] leader

SECTION 2: AGILE LEVELS

The 5 Agile Levels are comprehensive enough to depict the various states and levels most organizations could be at relative to agility?				✓	
The 5 Agile Levels are defined in a valid and logical order?					✓
One of our objectives is to make sure that the agile levels are practical and can be used in industry. In light of this, to what extent would you agree that these Agile Levels can be used to classify, and/or aid in the transition of, currently employed software development efforts?				✓	
The classification of agility into five Agile Levels as presented would be beneficial to the software engineering industry?					✓
Each of the Agile Levels presented encompass a set of agile practices and concepts. To what extent would you agree that those practices and concepts are relevant and correctly assigned to their respective agile levels?				✓	

Would you add, remove or redefine any of the 5 agile levels? If so please explain why.

No

Do you of any further comments about the Agile Levels?

OV03

SECTION 3: THE OVERALL PROCESS FRAMEWORK

	Strongly disagree	Slightly disagree	Neither disagree nor agree	Slightly agree	Strongly agree
[illegible]					
Overall objective of this process framework				×	
Discontinuing Factors					×
5 Agile Levels				×	
Project –Level Assessment				×	
Organizational Assessment				×	
Gap Analysis					×
[illegible]					
One of our objectives is to make sure that the process framework is practical and can be used in Industry. In light of this to what extent would you agree that process framework would be practical and feasible to employ?				✓	
[illegible]					
The process framework is beneficial to the software engineering industry?				×	
[illegible]					
All the necessary components are present in this process framework in order to achieve its overall goal of aiding an organization in the adoption of agile development for its various projects?				✓	
The steps and activities in the process framework are organized in a logical and valid sequence in order to achieve its overall goal of an assisting agile adoption?				✓	
[illegible]					
Discontinuing Factors				×	
5 Agile Levels					✓
Project –Level Assessment				×	
Organizational Assessment				✓	
Gap Analysis				×	

Would you add, remove or redefine any components of this process framework? If so please explain why.

Might look at strengthening "why" we're adopting given practices, and linkage b/w project - org level assessment. Also, when assessment takes place (before, after, during)

Do you have any additional comments about the process framework?

SECTION 5: FURTHER EVALUTION

[illegible]	YES	NO	[✓ MAYBE]

OV04

VIRGINIA POLYTECHNIC INSTITUTE AND STATE UNIVERSITY

Virginia Tech

Assessment Questionnaire:
"Process for Adoption of Agile Practices in Projects"

Date:	Reference # *(for archiving purposes)*:

SECTION 1: ASSESSOR'S INFORMATION

Name (Optional):

Organization / Institute:

Official Position: Marketing Mgr. **Years in position :** /

Email : **Phone Number :**

Please rate your familiarity with Agile software development	1 Not familiar	2	3	4 somewhat familiar	[✓ 5]	6	7 expert
Please rate your highest level of involvement in development efforts that used Agile practices	1 None	2	3	[✓ 4] participant	5	6	7 leader
How frequently have you aided entities in adopting Agile practices	1 Never	2	3	4 occasionally	[✓ 5]	6	7 constantly
Please rate your level of familiarity with general process assessment and/or process improvement	1 None	2	3	4 participant	5	[✓ 6]	7 leader

SECTION 2: AGILE LEVELS

The 5 Agile Levels are comprehensive enough to depict the various states and levels most organizations could be at relative to agility?					✓
The 5 Agile Levels are defined in a valid and logical order?					✓
One of our objectives is to make sure that the agile levels are practical and can be used in industry. In light of this, to what extent would you agree that these Agile Levels can be used to classify, and/or aid in the transition of, currently employed software development efforts?				✓	
The classification of agility into five Agile Levels as presented would be beneficial to the software engineering industry?				✓	
Each of the Agile Levels presented encompass a set of agile practices and concepts. To what extent would you agree that those practices and concepts are relevant and correctly assigned to their respective agile levels?				✓	

Would you add, remove or redefine any of the 5 agile levels? If so please explain why.

Agree w/earlier comment — customer + embracing change.

Do you have **of any further comments about the Agile Levels?**

Clarify the application + appropriate order of assessment between organization + project levels

0V04

SECTION 3: THE OVERALL PROCESS FRAMEWORK

	Strongly [illegible]	Slightly [illegible]	Neither [illegible]	Slightly [illegible]	Strongly [illegible]
[illegible]					
Overall objective of this process framework					✓
Discontinuing Factors					✓
5 Agile Levels				✓	
Project –Level Assessment					
Organizational Assessment			✓		
Gap Analysis				✓	
[illegible]					
One of our objectives is to make sure that the process framework is practical and can be used in industry. In light of this to what extent would you agree that process framework would be practical and feasible to employ?					✓
[illegible]					
The process framework is beneficial to the software engineering industry?				✓	
[illegible]					
All the necessary components are present in this process framework in order to achieve its overall goal of aiding an organization in the adoption of agile development for its various projects?				✓	
The steps and activities in the process framework are organized in a logical and valid sequence in order to achieve its overall goal of an assisting agile adoption?					✓
[illegible]					
Discontinuing Factors					✓
5 Agile Levels					✓
Project –Level Assessment					✓
Organizational Assessment					✓
Gap Analysis					✓

Would you add, remove or redefine any components of this process framework? If so please explain why.

Application to business side vs technical development unit. Allow for variables w/in methods as affecting overall appropriate application to assessment

Do you have any additional comments about the process framework?

SECTION 5: FURTHER EVALUTION

Would [illegible] research[illegible]	YES	NO	[✓ MAYBE]

OV05

VIRGINIA POLYTECHNIC INSTITUTE AND STATE UNIVERSITY

Virginia Tech

Assessment Questionnaire:
"Process for Adoption of Agile Practices in Projects"

Date:	Reference # *(for archiving purposes)*:

SECTION 1: ASSESSOR'S INFORMATION

Name (Optional):

Organization / Institute:

Official Position: MANAGING DIRECTOR	**Years in position :** 2
Email :	**Phone Number :**

Please rate your familiarity with Agile software development	1 Not familiar	2	3	4 somewhat familiar	5	6	[✓ 7 expert]
Please rate your highest level of involvement in development efforts that used Agile practices	1 None	2	3	4 participant	5	6	[✓ 7 leader]
How frequently have you aided entities in adopting Agile practices	1 Never	2	3	4 occasionally	5	6	[✓ 7 constantly]
Please rate your level of familiarity with general process assessment and/or process improvement	1 None	2	3	4 participant	5	6	[✓ 7 leader]

SECTION 2: AGILE LEVELS

The 5 Agile Levels are comprehensive enough to depict the various states and levels most organizations could be at relative to agility?					✓
The 5 Agile Levels are defined in a valid and logical order?					✓
One of our objectives is to make sure that the agile levels are practical and can be used in industry. In light of this, to what extent would you agree that these Agile Levels can be used to classify, and/or aid in the transition of, currently employed software development efforts?					✓
The classification of agility into five Agile Levels as presented would be beneficial to the software engineering industry?					✓
Each of the Agile Levels presented encompass a set of agile practices and concepts. To what extent would you agree that those practices and concepts are relevant and correctly assigned to their respective agile levels?					✓

Would you add, remove or redefine any of the 5 agile levels? If so please explain why.

ANALYSE JERRY WEINBERG'S LEVELS

Do you of any further comments about the Agile Levels?

OVØ5

SECTION 3: THE OVERALL PROCESS FRAMEWORK

	[illegible]	[illegible]	To some degree	[illegible]	[illegible]
[illegible]					
Overall objective of this process framework					✓
Discontinuing Factors					✓
5 Agile Levels					✓
Project –Level Assessment					✓
Organizational Assessment			✓		
Gap Analysis					✓
[illegible]					
One of our objectives is to make sure that the process framework is practical and can be used in industry. In light of this to what extent would you agree that process framework would be practical and feasible to employ?					✓
[illegible]					
The process framework is beneficial to the software engineering industry?					✓
[illegible]					
All the necessary components are present in this process framework in order to achieve its overall goal of aiding an organization in the adoption of agile development for its various projects?				✓	
The steps and activities in the process framework are organized in a logical and valid sequence in order to achieve its overall goal of an assisting agile adoption?				✓	
[illegible]					
Discontinuing Factors					✓
5 Agile Levels					✓
Project –Level Assessment					✓
Organizational Assessment					✓
Gap Analysis					✓

Would you add, remove or redefine any components of this process framework? If so please explain why.

Do you have any additional comments about the process framework?

SECTION 5: FURTHER EVALUTION

[illegible] research?	YES	NO	[✓ MAYBE]

VIRGINIA POLYTECHNIC INSTITUTE AND STATE UNIVERSITY

Virginia Tech

Assessment Questionnaire:
"Process for Adoption of Agile Practices in Projects"

OV06

Date:	Reference #: *(for archiving purposes):*

SECTION 1: ASSESSOR'S INFORMATION

Name (Optional): [redacted]

Organization / Institute: [redacted]

Official Position: SENIOR XP COACH **Years in position :** 6

Email : [redacted] **Phone Number :** [redacted]

Please rate your familiarity with Agile software development	1 Not familiar	2	3	4 somewhat familiar	5	6	[✓ 7] expert
Please rate your highest level of involvement in development efforts that used Agile practices	1 None	2	3	4 participant	5	6	[✓ 7] leader
How frequently have you aided entities in adopting Agile practices	1 Never	2	3	4 occasionally	5	6	[✓ 7] constantly
Please rate your level of familiarity with general process assessment and/or process improvement	1 None	2	3	4 participant	5	[✓ 6]	7 leader

SECTION 2: AGILE LEVELS

The 5 Agile Levels are comprehensive enough to depict the various states and levels most organizations could be at relative to agility?					✓
The 5 Agile Levels are defined in a valid and logical order?				✓	
One of our objectives is to make sure that the agile levels are practical and can be used in industry. In light of this, to what extent would you agree that these Agile Levels can be used to classify, and/or aid in the transition of, currently employed software development efforts?					✓
The classification of agility into five Agile Levels as presented would be needed and beneficial to the software engineering industry?					✓
Each of the Agile Levels presented encompass a set of agile practices and concepts. To what extent would you agree that those practices and concepts are relevant and correctly assigned to their respective agile levels?					✓

Would you add, remove or redefine any of the 5 agile levels? If so please explain why.

"Level" itself may need to be replaced. "Dimension", "region", "focus"?

I find that I can start working on, say, TDD, even while I am blocked on customer involvement.

Do you of any further comments about the Agile Levels?

OVØ6

SECTION 3: THE OVERALL PROCESS FRAMEWORK

	strongly disagree	slightly disagree	neither disagree nor agree	slightly agree	strongly agree
UNDERSTANDABILITY					
[illegible]					
Overall objective of this process framework					✓
Discontinuing Factors					✓ ★ ☆
5 Agile Levels					✓
Project –Level Assessment					✓
Organizational Assessment					✓
Gap Analysis					✓
PRACTICALITY					
One of our objectives is to make sure that the process framework is practical and can be used in industry. In light of this to what extent would you agree that process framework would be practical and feasible to employ?					✓
NECESSITY					
The process framework is needed and beneficiary in the software engineering industry?					✓
COMPLETENESS					
All the necessary components are present in this process framework in order to achieve its overall goal of aiding an organization in the adoption of agile development for its various projects?				✓	
The steps and activities in the process framework are organized in a logical and valid sequence in order to achieve its overall goal of an assisting agile adoption?					✓
[illegible]					
Discontinuing Factors					✓
5 Agile Levels				✓	
Project –Level Assessment					✓
Organizational Assessment					✓
Gap Analysis					✓

Would you add, remove or redefine any components of this process framework? If so please explain why.

Do you have any additional comments about the process framework?

★ – transparency seems irremediable. It stops more shows than the other DF's.

Overall, this is first-rate work. I commend Ahmed's efforts and endorse his work wholeheartedly as legitimate in its interest and merit to our industry.

SECTION 5: FURTHER EVALUTION

Would you like to participate in an extensive, more detailed evaluation of this research?	[✓ YES]	NO	MAYBE

VIRGINIA POLYTECHNIC INSTITUTE AND STATE UNIVERSITY

Virginia Tech

Assessment Questionnaire:
"Process for Adoption of Agile Practices in Projects"

OVO+

Date: Oct 11, 2006	Reference # *(for archiving purposes)*:						
SECTION 1: ASSESSOR'S INFORMATION							
Name (Optional): [redacted]							
Organization / Institute: [redacted]							
Official Position: President	**Years in position :** 12						
Email : [redacted]	**Phone Number :** [redacted]						
Please rate your familiarity with Agile software development	1 Not familiar	2	3	4 somewhat familiar	5	6	[✓ 7 expert]
Please rate your highest level of involvement in development efforts that used Agile practices	1 None	2	3	4 participant	5	6	[✓ 7 leader]
How frequently have you aided entities in adopting Agile practices	1 Never	2	3	4 occasionally	5	6	[✓ 7 constantly]
Please rate your level of familiarity with general process assessment and/or process improvement	1 None	2	3	4 participant	5	6	[✓ 7 leader]

SECTION 2: AGILE LEVELS

The 5 Agile Levels are comprehensive enough to depict the various states and levels most organizations could be at relative to agility?					✓
The 5 Agile Levels are defined in a valid and logical order?					✓
One of our objectives is to make sure that the agile levels are practical and can be used in industry. In light of this, to what extent would you agree that these Agile Levels can be used to classify, and/or aid in the transition of, currently employed software development efforts?					✓
The classification of agility into five Agile Levels as presented would be beneficial to the software engineering industry?					✓
Each of the Agile Levels presented encompass a set of agile practices and concepts. To what extent would you agree that those practices and concepts are relevant and correctly assigned to their respective agile levels?				✓	

Would you add, remove or redefine any of the 5 agile levels? If so please explain why.

I'm sure I'd move a few of the practices around but not in a significant way. I love the 5 labels for the levels.

Do you of any further comments about the Agile Levels?

OV07

SECTION 3: THE OVERALL PROCESS FRAMEWORK

	[illegible]	[illegible]	[illegible]	[illegible]	Strongly Agree
[illegible]					
Overall objective of this process framework					✓
Discontinuing Factors					✓
5 Agile Levels					✓
Project –Level Assessment					✓
Organizational Assessment					✓
Gap Analysis					✓
[illegible]					
One of our objectives is to make sure that the process framework is practical and can be used in industry. In light of this to what extent would you agree that process framework would be practical and feasible to employ?					✓
[illegible]					
The process framework is beneficial to the software engineering industry?					✓
[illegible]					
All the necessary components are present in this process framework in order to achieve its overall goal of aiding an organization in the adoption of agile development for its various projects?					✓
The steps and activities in the process framework are organized in a logical and valid sequence in order to achieve its overall goal of an assisting agile adoption?					✓
[illegible]					
Discontinuing Factors				✓	
5 Agile Levels					✓
Project –Level Assessment					✓
Organizational Assessment					✓
Gap Analysis					✓

Would you add, remove or redefine any components of this process framework? If so please explain why.

No

Do you have any additional comments about the process framework?

I think this is fantastic.

SECTION 5: FURTHER EVALUTION

[illegible] research?	[✓ YES]	NO	MAYBE

CV48

VIRGINIA POLYTECHNIC INSTITUTE AND STATE UNIVERSITY

Virginia Tech

Assessment Questionnaire:
"Process for Adoption of Agile Practices in Projects"

Date: 3 August 2006 | Reference # *(for archiving purposes)*:

SECTION 1: ASSESSOR'S INFORMATION

Name (Optional):

Organization / Institute:

Official Position: CHIEF SCIENTIST **Years in position :** 3

Email : **Phone Number :**

Please rate your familiarity with Agile software development	1 Not familiar	2	3	4 somewhat familiar	5	6	[✓ 7] expert
Please rate your highest level of involvement in development efforts that used Agile practices	1 None	2	3	4 participant	5	6	[✓ 7] leader
How frequently have you aided entities in adopting Agile practices	1 Never	2	3	4 occasionally	5	6	[✓ 7] constantly
Please rate your level of familiarity with general process assessment and/or process improvement	1 None	2	3	4 participant	[✓ 5]	6	7 leader

SECTION 2: AGILE LEVELS

	strongly disagree	slightly disagree	neither disagree nor agree	slightly agree	strongly agree
COMPREHENSIVENESS					
The 5 Agile Levels are comprehensive enough to depict the various states and levels most organizations could be at relative to agility?		✓			
The 5 Agile Levels are defined in a valid and logical order?			✓		
PRACTICALITY					
One of our objectives is to make sure that the agile levels are practical and can be used in industry. In light of this, to what extent would you agree that these Agile Levels can be used to classify, and/or aid in the transition of, currently employed software development efforts?		✓			
NECESSITY					
The classification of agility into five Agile Levels as presented would be needed and beneficial to the software engineering industry?					✓
RELEVANCE					
Each of the Agile Levels presented encompass a set of agile practices and concepts. To what extent would you agree that those practices and concepts are relevant and correctly assigned to their respective agile levels?	✓				

Would you add, remove or redefine any of the 5 agile levels? If so please explain why.

I'm uncomfortable with the specifics of the levels. It's very difficult when you start talking about specific practices.

Do you of any further comments about the Agile Levels?

This is where the devil is in the details. For example, it's very hard to do effective iterative development without something like user stories. ~~[illegible]~~ Sitting together also seems to be very important to get beyond a basic level of agility.

SECTION 3: THE OVERALL PROCESS FRAMEWORK

	strongly disagree	slightly disagree	neither disagree nor agree	slightly agree	strongly agree
UNDERSTANDABILITY					
For the topics listed below designate the degree to which you agree that they are understandable					
Overall objective of this process framework					✓
Discontinuing Factors					✓
5 Agile Levels					✓
Project –Level Assessment			✓		
Organizational Assessment			✓		
Gap Analysis		✓			
PRACTICALITY					
One of our objectives is to make sure that the process framework is practical and can be used in industry. In light of this to what extent would you agree that process framework would be practical and feasible to employ?			✓		
NECESSITY					
The process framework is needed and beneficiary in the software engineering industry?				✓	
COMPLETENESS					
All the necessary components are present in this process framework in order to achieve its overall goal of aiding an organization in the adoption of agile development for its various projects?			✓		
The steps and activities in the process framework are organized in a logical and valid sequence in order to achieve its overall goal of an assisting agile adoption?				✓	
EFFECTIVENESS					
To what extent do you agree that each of the components in this process framework is necessary and sufficient for the framework to achieve its purpose?					
Discontinuing Factors					✓
5 Agile Levels					✓
Project –Level Assessment					✓
Organizational Assessment					✓
Gap Analysis				✓	

Would you add, remove or redefine any components of this process framework? If so please explain why.

framework seems fine.

Do you have any additional comments about the process framework?

Are you working with an expert in organizational change — someone on the committee?

SECTION 5: FURTHER EVALUTION

Would you like to participate in an extensive, more detailed, evaluation of this research?	[✓ YES]	NO	MAYBE

VIRGINIA POLYTECHNIC INSTITUTE AND STATE UNIVERSITY

Virginia Tech

Assessment Questionnaire:
"Process for Adoption of Agile Practices in Projects"

Over

Date:	Reference # *(for archiving purposes):*

SECTION 1: ASSESSOR'S INFORMATION

Name (Optional):

Organization / Institute:

Official Position: Agile Practice lead **Years in position :** 2

Email : **Phone Number :**

Please rate your familiarity with Agile software development	1 Not familiar	2	3	4 somewhat familiar	5	6	[✓ 7] expert
Please rate your highest level of involvement in development efforts that used Agile practices	1 None	2	3	4 participant	5	6	[✓ 7] leader
How frequently have you aided entities in adopting Agile practices	1 Never	2	3	4 occasionally	5	6	[✓ 7] constantly
Please rate your level of familiarity with general process assessment and/or process improvement	1 None	2	3	4 participant	5	6	[✓ 7] leader

SECTION 2: AGILE LEVELS

	strongly disagree	slightly disagree	neither disagree nor agree	slightly agree	strongly agree
COMPREHENSIVENESS					
The 5 Agile Levels are comprehensive enough to depict the various states and levels most organizations could be at relative to agility?				✓	
The 5 Agile Levels are defined in a valid and logical order?					✓
PRACTICALITY					
One of our objectives is to make sure that the agile levels are practical and can be used in industry. In light of this, to what extent would you agree that these Agile Levels can be used to classify, and/or aid in the transition of, currently employed software development efforts?				✓	
NECESSITY					
The classification of agility into five Agile Levels as presented would be needed and beneficial to the software engineering industry?				✓	
RELEVANCE					
Each of the Agile Levels presented encompass a set of agile practices and concepts. To what extent would you agree that those practices and concepts are relevant and correctly assigned to their respective agile levels?		✓			

Challenged to define the practice -- seems there is a range within the practice that span levels.

Would you add, remove or redefine any of the 5 agile levels? If so please explain why.

Do you of any further comments about the Agile Levels?

levels are good -- but practices seem better as guidance rather than law?

SECTION 3: THE OVERALL PROCESS FRAMEWORK

	strongly disagree	slightly disagree	neither disagree nor agree	slightly agree	strongly agree
UNDERSTANDABILITY					
For the topics listed below designate the degree to which you agree that they are understandable					
Overall objective of this process framework					✓
Discontinuing Factors			✓		
5 Agile Levels				✓	
Project –Level Assessment				✓	
Organizational Assessment			✓		
Gap Analysis				✓	
PRACTICALITY					
One of our objectives is to make sure that the process framework is practical and can be used in industry. In light of this to what extent would you agree that process framework would be practical and feasible to employ?					
NECESSITY					
The process framework is needed and beneficiary in the software engineering industry?			✓		
COMPLETENESS					
All the necessary components are present in this process framework in order to achieve its overall goal of aiding an organization in the adoption of agile development for its various projects?				✓	
The steps and activities in the process framework are organized in a logical and valid sequence in order to achieve its overall goal of an assisting agile adoption?				✓	
EFFECTIVENESS					
To what extent do you agree that each of the components in this process framework is necessary and sufficient for the framework to achieve its purpose?					
Discontinuing Factors					✓
5 Agile Levels					✓
Project –Level Assessment				✓	
Organizational Assessment				✓	
Gap Analysis			✓		

Over

Would you add, remove or redefine any components of this process framework? If so please explain why.

practices should be guidance. Vs. fixed.

Do you have any additional comments about the process framework?

- Concern about limiting the organizations potential
- What does organizational change intersect w/ framework

SECTION 5: FURTHER EVALUTION

Would you like to participate in an extensive, more detailed, evaluation of this research?	[✓ **YES**]	**NO**	**MAYBE**

VIRGINIA POLYTECHNIC INSTITUTE AND STATE UNIVERSITY

Virginia Tech

Assessment Questionnaire:
"Process for Adoption of Agile Practices in Projects"

0V10

Date: 8/2/06	Reference # *(for archiving purposes):*						
SECTION 1: ASSESSOR'S INFORMATION							
Name (Optional):							
Organization / Institute:							
Official Position: SVP	**Years in position :** 5						
Email :	**Phone Number :**						
Please rate your familiarity with Agile software development	1 Not familiar	2	3	4 somewhat familiar	5	[✓ 6]	7 expert
Please rate your highest level of involvement in development efforts that used Agile practices	1 None	2	3	4 participant	5	6	[✓ 7] leader
How frequently have you aided entities in adopting Agile practices	1 Never	2	3	4 occasionally	5	[✓ 6]	7 constantly
Please rate your level of familiarity with general process assessment and/or process improvement	1 None	2	3	4 participant	5	[✓ 6]	7 leader

SECTION 2: AGILE LEVELS

	strongly disagree	slightly disagree	neither disagree nor agree	slightly agree	strongly agree
COMPREHENSIVENESS					
The 5 Agile Levels are comprehensive enough to depict the various states and levels most organizations could be at relative to agility?		✓			
The 5 Agile Levels are defined in a valid and logical order?				✓	
PRACTICALITY					
One of our objectives is to make sure that the agile levels are practical and can be used in industry. In light of this, to what extent would you agree that these Agile Levels can be used to classify, and/or aid in the transition of, currently employed software development efforts?				✓	
NECESSITY					
The classification of agility into five Agile Levels as presented would be needed and beneficial to the software engineering industry?					✓
RELEVANCE					
Each of the Agile Levels presented encompass a set of agile practices and concepts. To what extent would you agree that those practices and concepts are relevant and correctly assigned to their respective agile levels?		✓			

Would you add, remove or redefine any of the 5 agile levels? If so please explain why.

I agree with the 5 levels and the themes for each level.
The principles look to be correct.
I do not know that the characteristics for each
Level / Process are correct. Examples would
be VERY helpful.

Do you of any further comments about the Agile Levels?

0710

SECTION 3: THE OVERALL PROCESS FRAMEWORK

	strongly disagree	slightly disagree	neither disagree nor agree	slightly agree	strongly agree
UNDERSTANDABILITY					
For the topics listed below designate the degree to which you agree that they are understandable					
Overall objective of this process framework					✓
Discontinuing Factors				✓	
5 Agile Levels				✓	
Project –Level Assessment				✓	
Organizational Assessment			✓		
Gap Analysis			✓		
PRACTICALITY					
One of our objectives is to make sure that the process framework is practical and can be used in industry. In light of this to what extent would you agree that process framework would be practical and feasible to employ?				✓	
NECESSITY					
The process framework is needed and beneficiary in the software engineering industry?				✓	
COMPLETENESS					
All the necessary components are present in this process framework in order to achieve its overall goal of aiding an organization in the adoption of agile development for its various projects?	✓				
The steps and activities in the process framework are organized in a logical and valid sequence in order to achieve its overall goal of an assisting agile adoption?					✓
EFFECTIVENESS					
To what extent do you agree that each of the components in this process framework is necessary and sufficient for the framework to achieve its purpose?					
Discontinuing Factors				✓	
5 Agile Levels		✓			
Project –Level Assessment			✓		
Organizational Assessment			✓		
Gap Analysis			✓		

Would you add, remove or redefine any components of this process framework? If so please explain why.

Again, much more work is needed in the details of the 5 Agile levels – the questions & characteristics/practices.

Do you have any additional comments about the process framework?

SECTION 5: FURTHER EVALUTION

Would you like to participate in an extensive, more detailed, evaluation of this research?	**YES**	**NO**	[✓ **MAYBE**]

VIRGINIA POLYTECHNIC INSTITUTE AND STATE UNIVERSITY

Virginia Tech

Assessment Questionnaire:
"Process for Adoption of Agile Practices in Projects"

OV11

Date: 11/15/06	Reference # *(for archiving purposes)*:

SECTION 1: ASSESSOR'S INFORMATION

Name (Optional): [redacted]

Organization / Institute: [redacted]

Official Position: Founder **Years in position :** 11

Email : [redacted] **Phone Number :** [redacted]

Please rate your familiarity with Agile software development	1 Not familiar	2	3	4 somewhat familiar	5	6	[✓ 7 expert]
Please rate your highest level of involvement in development efforts that used Agile practices	1 None	2	3	4 participant	5	6	[✓ 7 leader]
How frequently have you aided entities in adopting Agile practices	1 Never	2	3	4 occasionally	5	6	[✓ 7 constantly]
Please rate your level of familiarity with general process assessment and/or process improvement	1 None	2	3	4 participant	5	[✓ 6]	7 leader

SECTION 2: AGILE LEVELS

The 5 Agile Levels are comprehensive enough to depict the various states and levels most organizations could be at relative to agility?			✓		
The 5 Agile Levels are defined in a valid and logical order?		✓			
One of our objectives is to make sure that the agile levels are practical and can be used in industry. In light of this, to what extent would you agree that these Agile Levels can be used to classify, and/or aid in the transition of, currently employed software development efforts?				✓	
The classification of agility into five Agile Levels as presented would be beneficial to the software engineering industry?		✓			
Each of the Agile Levels presented encompass a set of agile practices and concepts. To what extent would you agree that those practices and concepts are relevant and correctly assigned to their respective agile levels?	✓				

Would you add, remove or redefine any of the 5 agile levels? If so please explain why.

I would not name the levels.

Do you of any further comments about the Agile Levels?

I like the ~~[illegible]~~ Measurement Index
– I think "Agile Levels" is a slippery
slope.

SECTION 3: THE OVERALL PROCESS FRAMEWORK

	[illegible]				
[illegible]					
Overall objective of this process framework					✓
Discontinuing Factors					✓
5 Agile Levels		✓			
Project –Level Assessment		✓			
Organizational Assessment					✓
Gap Analysis					✓
[illegible]					
One of our objectives is to make sure that the process framework is practical and can be used in industry. In light of this to what extent would you agree that process framework would be practical and feasible to employ?				✓	
[illegible]					
The process framework is beneficial to the software engineering industry?					✓
[illegible]					
All the necessary components are present in this process framework in order to achieve its overall goal of aiding an organization in the adoption of agile development for its various projects?				✓	
The steps and activities in the process framework are organized in a logical and valid sequence in order to achieve its overall goal of an assisting agile adoption?					✓
[illegible]					
Discontinuing Factors					✓
5 Agile Levels				✓	
Project –Level Assessment				✓	
Organizational Assessment					✓
Gap Analysis					✓

Would you add, remove or redefine any components of this process framework? If so please explain why.

Do you have any additional comments about the process framework?

SECTION 5: FURTHER EVALUTION

[illegible] research?	YES	NO	[✓ MAYBE]

OV12

VIRGINIA POLYTECHNIC INSTITUTE AND STATE UNIVERSITY

Virginia Tech

Assessment Questionnaire:
"Process for Adoption of Agile Practices in Projects"

Date: | Reference # *(for archiving purposes)*:

SECTION 1: ASSESSOR'S INFORMATION

Name (Optional): [redacted]

Organization / Institute: [redacted]

Official Position: Senior XP Coach | **Years in position :**

Email : [redacted] | **Phone Number :**

Please rate your familiarity with Agile software development	1 Not familiar	2	3	4 somewhat familiar	5	[✓ 6]	7 expert
Please rate your highest level of involvement in development efforts that used Agile practices	1 None	2	3	4 participant	5	6	[✓ 7] leader
How frequently have you aided entities in adopting Agile practices	1 Never	2	3	4 occasionally	5	[✓ 6]	7 constantly
Please rate your level of familiarity with general process assessment and/or process improvement	1 None	2	3	4 participant	[✓ 5]	6	7 leader

SECTION 2: AGILE LEVELS

[illegible]					
The 5 Agile Levels are comprehensive enough to depict the various states and levels most organizations could be at relative to agility?		✓			
The 5 Agile Levels are defined in a valid and logical order?		✓			
[illegible]					
One of our objectives is to make sure that the agile levels are practical and can be used in industry. In light of this, to what extent would you agree that these Agile Levels can be used to classify, and/or aid in the transition of, currently employed software development efforts?		✓			
[illegible]					
The classification of agility into five Agile Levels as presented would be beneficial to the software engineering industry?		✓			
[illegible]					
Each of the Agile Levels presented encompass a set of agile practices and concepts. To what extent would you agree that those practices and concepts are relevant and correctly assigned to their respective agile levels?	✓				

Would you add, remove or redefine any of the 5 agile levels? If so please explain why.

Probably wouldn't use levels

Do you of any further comments about the Agile Levels?

SECTION 3: THE OVERALL PROCESS FRAMEWORK

	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]					
Overall objective of this process framework					✓
Discontinuing Factors					
5 Agile Levels				✓	
Project –Level Assessment					✓
Organizational Assessment				✓	
Gap Analysis				✓	
[illegible]					
One of our objectives is to make sure that the process framework is practical and can be used in industry. In light of this to what extent would you agree that process framework would be practical and feasible to employ?				✓	
[illegible]					
The process framework is beneficial to the software engineering industry?					✓
[illegible]					
All the necessary components are present in this process framework in order to achieve its overall goal of aiding an organization in the adoption of agile development for its various projects?				✓	
The steps and activities in the process framework are organized in a logical and valid sequence in order to achieve its overall goal of an assisting agile adoption?					✓
[illegible]					
Discontinuing Factors				✓	
5 Agile Levels		✓			
Project –Level Assessment				✓	
Organizational Assessment					✓
Gap Analysis			✓		

Would you add, remove or redefine any components of this process framework? If so please explain why.

We would probably do something differen than level 5 to assess which practices we want to assess for readiness and we would probably make it client-specific

Do you have any additional comments about the process framework?

SECTION 5: FURTHER EVALUTION

[illegible] research?	YES	NO	[✓ MAYBE]

VIRGINIA POLYTECHNIC INSTITUTE AND STATE UNIVERSITY

Virginia Tech

Assessment Questionnaire:
"Process for Adoption of Agile Practices in Projects"

CV15

Date: 11.3.2006	Reference # *(for archiving purposes)*:

SECTION 1: ASSESSOR'S INFORMATION

Name (Optional): [redacted]

Organization / Institute: [redacted]

Official Position: PROG/ANALYST **Years in position :** —

Email : [redacted] **Phone Number :** [redacted]

Please rate your familiarity with Agile software development	1 Not familiar	2	3	4 somewhat familiar	[✓ 5]	6	7 expert
Please rate your highest level of involvement in development efforts that used Agile practices	1 None	2	[✓ 3]	4 participant	5	6	7 leader
How frequently have you aided entities in adopting Agile practices	1 Never	2	3	[✓ 4] occasionally	5	6	7 constantly
Please rate your level of familiarity with general process assessment and/or process improvement	1 None	2	3	[✓ 4] participant	5	6	7 leader

SECTION 2: AGILE LEVELS

The 5 Agile Levels are comprehensive enough to depict the various states and levels most organizations could be at relative to agility?				X	
The 5 Agile Levels are defined in a valid and logical order?			X		
One of our objectives is to make sure that the agile levels are practical and can be used in industry. In light of this, to what extent would you agree that these Agile Levels can be used to classify, and/or aid in the transition of, currently employed software development efforts?				X	
The classification of agility into five Agile Levels as presented would be beneficial to the software engineering industry?				X	
Each of the Agile Levels presented encompass a set of agile practices and concepts. To what extent would you agree that those practices and concepts are relevant and correctly assigned to their respective agile levels?			X		

Would you add, remove or redefine any of the 5 agile levels? If so please explain why.

Do you of any further comments about the Agile Levels?

CV15

SECTION 3: THE OVERALL PROCESS FRAMEWORK

	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]					
Overall objective of this process framework					X
Discontinuing Factors					X
5 Agile Levels			X		
Project –Level Assessment				X	
Organizational Assessment			X		
Gap Analysis				X	
[illegible]					
One of our objectives is to make sure that the process framework is practical and can be used in industry. In light of this to what extent would you agree that process framework would be practical and feasible to employ?				X	
[illegible]					
The process framework is beneficial to the software engineering industry?				X	
[illegible]					
All the necessary components are present in this process framework in order to achieve its overall goal of aiding an organization in the adoption of agile development for its various projects?					X
The steps and activities in the process framework are organized in a logical and valid sequence in order to achieve its overall goal of an assisting agile adoption?					X
[illegible]					
Discontinuing Factors				X	
5 Agile Levels				X	
Project –Level Assessment					X
Organizational Assessment					X
Gap Analysis					X

Would you add, remove or redefine any components of this process framework? If so please explain why.

Do you have any additional comments about the process framework?

SECTION 5: FURTHER EVALUTION

[illegible]	[✓ YES]	NO	MAYBE

VIRGINIA POLYTECHNIC INSTITUTE AND STATE UNIVERSITY

Virginia Tech

Assessment Questionnaire:
"Process for Adoption of Agile Practices in Projects"

0V14

Date: 11/2/06	Reference # *(for archiving purposes)*:						
SECTION 1: ASSESSOR'S INFORMATION							
Name (Optional) [redacted]							
Organization / Institute: [redacted]							
Official Position: PRESIDENT	**Years in position :** 12						
Email : [redacted]	**Phone Number :** [redacted]						
Please rate your familiarity with Agile software development	1 Not familiar	2	3	4 somewhat familiar	5	6	[✓ 7] expert
Please rate your highest level of involvement in development efforts that used Agile practices	1 None	2	3	4 participant	5	6	[✓ 7] leader
How frequently have you aided entities in adopting Agile practices	1 Never	2	3	4 occasionally	5	6	[✓ 7] constantly
Please rate your level of familiarity with general process assessment and/or process improvement	1 None	2	3	4 participant	5	6	[✓ 7] leader

SECTION 2: AGILE LEVELS

COMPREHENSIVENESS					
The 5 Agile Levels are comprehensive enough to depict the various states and levels most organizations could be at relative to agility?					✓
The 5 Agile Levels are defined in a valid and logical order?				✓	
PRACTICALITY					
One of our objectives is to make sure that the agile levels are practical and can be used in industry. In light of this, to what extent would you agree that these Agile Levels can be used to classify, and/or aid in the transition of, currently employed software development efforts?					✓
NECESSITY					
The classification of agility into five Agile Levels as presented would be beneficial to the software engineering industry?				✓	
RELEVANCE					
Each of the Agile Levels presented encompass a set of agile practices and concepts. To what extent would you agree that those practices and concepts are relevant and correctly assigned to their respective agile levels?			✓		

Would you add, remove or redefine any of the 5 agile levels? If so please explain why.

Do you of any further comments about the Agile Levels?

OV/14

SECTION 3: THE OVERALL PROCESS FRAMEWORK

	Strongly disagree	Slightly disagree	Neither disagree nor agree	Slightly agree	Strongly agree
UNDERSTANDABILITY [illegible]					
Overall objective of this process framework					✓
Discontinuing Factors					✓
5 Agile Levels				✓	
Project –Level Assessment					✓
Organizational Assessment					✓
Gap Analysis					✓
PRACTICALITY					
One of our objectives is to make sure that the process framework is practical and can be used in industry. In light of this to what extent would you agree that process framework would be practical and feasible to employ?					✓
USEFULNESS					
The process framework is beneficial to the software engineering industry?					✓
COMPLETENESS					
All the necessary components are present in this process framework in order to achieve its overall goal of aiding an organization in the adoption of agile development for its various projects?			✓		
The steps and activities in the process framework are organized in a logical and valid sequence in order to achieve its overall goal of an assisting agile adoption?				✓	
[illegible]					
Discontinuing Factors					✓
5 Agile Levels			✓		
Project –Level Assessment				✓	
Organizational Assessment					✓
Gap Analysis					✓

Would you add, remove or redefine any components of this process framework? If so please explain why.

Do you have any additional comments about the process framework?

SECTION 5: FURTHER EVALUTION

Would you like to participate in a more in-depth evaluation of this research?	[✓ YES] !	NO	MAYBE

VIRGINIA POLYTECHNIC INSTITUTE AND STATE UNIVERSITY

Virginia Tech

Assessment Questionnaire: "Process for Adoption of Agile Practices in Projects"

OV15

Date: 11/2/06

Reference # *(for archiving purposes)*:

SECTION 1: ASSESSOR'S INFORMATION

Name (Optional): [redacted]

Organization / Institute: [redacted]

Official Position: Architect, Interaction Design

Years in position: 2.5

Email: [redacted]

Phone Number: [redacted]

Please rate your familiarity with Agile software development: 1 Not familiar, 2, 3, 4 somewhat familiar, 5, [✓ 6], 7 expert

Please rate your highest level of involvement in development efforts that used Agile practices: 1 None, 2, 3, 4 participant, 5, 6, [✓ 7 leader]

How frequently have you aided entities in adopting Agile practices: 1 Never, 2, 3, 4 occasionally, 5, 6, [✓ 7 constantly]

Please rate your level of familiarity with general process assessment and/or process improvement: 1 None, 2, 3, 4 participant, [✓ 5], 6, 7 leader

SECTION 2: AGILE LEVELS

The 5 Agile Levels are comprehensive enough to depict the various states and levels most organizations could be at relative to agility?		X			
The 5 Agile Levels are defined in a valid and logical order?				X	
One of our objectives is to make sure that the agile levels are practical and can be used in industry. In light of this, to what extent would you agree that these Agile Levels can be used to classify, and/or aid in the transition of, currently employed software development efforts?			X		
The classification of agility into five Agile Levels as presented would be beneficial to the software engineering industry?				X	
Each of the Agile Levels presented encompass a set of agile practices and concepts. To what extent would you agree that those practices and concepts are relevant and correctly assigned to their respective agile levels?		X			

Would you add, remove or redefine any of the 5 agile levels? If so please explain why.

Communication as a first level seems difficult – in practice I find this much harder to achieve than the adoption of techniques such as automated unit & acceptance testing, or estimation, or iterative/timeboxed development.

Do you of any further comments about the Agile Levels?

I'd like to see research somehow supporting this progression.

I'd like to understand better how levels would be affected as new techniques are discovered & incorporated.

OV15

SECTION 3: THE OVERALL PROCESS FRAMEWORK

	strongly disagree	slightly disagree	neither disagree nor agree	slightly agree	strongly agree
[illegible]					
Overall objective of this process framework					X
Discontinuing Factors					X
5 Agile Levels			X		
Project –Level Assessment					X
Organizational Assessment					X
Gap Analysis					X
[illegible]					
One of our objectives is to make sure that the process framework is practical and can be used in industry. In light of this to what extent would you agree that process framework would be practical and feasible to employ?					X
[illegible]					
The process framework is beneficial to the software engineering industry?					X
[illegible]					
All the necessary components are present in this process framework in order to achieve its overall goal of aiding an organization in the adoption of agile development for its various projects?			X		
The steps and activities in the process framework are organized in a logical and valid sequence in order to achieve its overall goal of an assisting agile adoption?					X
[illegible]					
Discontinuing Factors				X	
5 Agile Levels			X		
Project –Level Assessment					X
Organizational Assessment					X
Gap Analysis				X	

Would you add, remove or redefine any components of this process framework? If so please explain why.

- not without testing the framework through use

Do you have any additional comments about the process framework?

I'd like to test the framework - see how it holds up in use. As with all plans/processes/approaches, they may change as a result of real world application.

SECTION 5: FURTHER EVALUTION

Would you like to participate in [illegible] detailed evaluation [illegible] research? [✓ YES] NO [✓ MAYBE]

UV16

VIRGINIA POLYTECHNIC INSTITUTE AND STATE UNIVERSITY

Virginia Tech

Assessment Questionnaire:
"Process for Adoption of Agile Practices in Projects"

Date: NOVEMBER 1, 2006	Reference # *(for archiving purposes)*:

SECTION 1: ASSESSOR'S INFORMATION

Name (Optional): [redacted]

Organization / Institute: [redacted]

Official Position: OWNER / FOUNDER	**Years in position :** SEVERAL
Email [redacted]	**Phone Number :** [redacted]

Please rate your familiarity with Agile software development	1 Not familiar	2	3	4 somewhat familiar	5	6	[✓ 7 expert]
Please rate your highest level of involvement in development efforts that used Agile practices	1 None	2	3	4 participant	5	6	[✓ 7 leader]
How frequently have you aided entities in adopting Agile practices	1 Never	2	3	4 occasionally	[✓ 5]	6	7 constantly
Please rate your level of familiarity with general process assessement and/or process improvement	1 None	2	3	4 participant	5	[✓ 6]	7 leader

SECTION 2: AGILE LEVELS

[illegible]					
The 5 Agile Levels are comprehensive enough to depict the various states and levels most organizations could be at relative to agility?				✓	
The 5 Agile Levels are defined in a valid and logical order?				✓	
[illegible]					
One of our objectives is to make sure that the agile levels are practical and can be used in industry. In light of this, to what extent would you agree that these Agile Levels can be used to classify, and/or aid in the transition of, currently employed software development efforts?				✓	
[illegible]					
The classification of agility into five Agile Levels as presented would be beneficial to the software engineering industry?				✓	
[illegible]					
Each of the Agile Levels presented encompass a set of agile practices and concepts. To what extent would you agree that those practices and concepts are relevant and correctly assigned to their respective agile levels?				✓	

Would you add, remove or redefine any of the 5 agile levels? If so please explain why.

Do you of any further comments about the Agile Levels?

AS WE DISCUSSED, THE LEVELS NEED TO BE INTERCHANGEABLE WITH OTHER ASSESMENT METHODOLOGIES. I THINK YOU CURRENTLY HAVE TOO MUCH FOCUS ON THE LEVELS. EXPLAIN THE FRAMEWORK WITH A GENERIC STEP FOR ASSESSMENT. THEN COME BACK AROUND AND EXPLAIN YOUR PROPOSED ASSESSMENT METHOD.

UV10

SECTION 3: THE OVERALL PROCESS FRAMEWORK

	strongly disagree	slightly disagree	neither disagree nor agree	slightly agree	strongly agree
[illegible]					
Overall objective of this process framework				✓	
Discontinuing Factors					✓
5 Agile Levels				✓	
Project –Level Assessment					✓
Organizational Assessment					✓
Gap Analysis					✓
[illegible]					
One of our objectives is to make sure that the process framework is practical and can be used in industry. In light of this to what extent would you agree that process framework would be practical and feasible to employ?				✓	
[illegible]					
The process framework is beneficial to the software engineering industry?				✓	
[illegible]					
All the necessary components are present in this process framework in order to achieve its overall goal of aiding an organization in the adoption of agile development for its various projects?		✓			
The steps and activities in the process framework are organized in a logical and valid sequence in order to achieve its overall goal of an assisting agile adoption?			✓		
[illegible]					
Discontinuing Factors					✓
5 Agile Levels				✓	
Project –Level Assessment					✓
Organizational Assessment					✓
Gap Analysis					✓

Would you add, remove or redefine any components of this process framework? If so please explain why.

SOMETHING IS BUGGING ME ABOUT THE TITLE. IT IS SUPPOSED TO BE A FRAMEWORK FOR ADOPTING AGILE. BUT IT SEEMS TO BE A FRAMEWORK FOR ASSESSING READINESS. THE LAST STEP IN THE PROCESS IS "PROCEED TO THE ADOPTION..." HOW IS THAT HELPING ME ADOPT AGILE. IT IS LIKE YOU TOLD ME I CAN TAKE MY PROJECT TO A LEVEL 4, AND TOLD ME TO JUST RUN OFF AND MAKE IT HAPPEN!

Do you have any additional comments about the process framework?

SECTION 5: FURTHER EVALUTION

Would [illegible] research?	YES	NO	[✓ MAYBE]

OV17

VIRGINIA POLYTECHNIC INSTITUTE AND STATE UNIVERSITY

Virginia Tech

Assessment Questionnaire:
"Process for Adoption of Agile Practices in Projects"

Date: | Reference # *(for archiving purposes)*:

SECTION 1: ASSESSOR'S INFORMATION

Name (Optional): [redacted]

Organization / Institute: [redacted]

Official Position: Independent consultant | **Years in position :** 0. Previously Agile Test Manager/5 yrs
Agile Dev Lead/2 yrs

Email [redacted] | **Phone Number :**

Please rate your familiarity with Agile software development	1 Not familiar	2	3	4 somewhat familiar	5	[✓ 6]	7 expert
Please rate your highest level of involvement in development efforts that used Agile practices	1 None	2	3	4 participant	5	6	[✓ 7] leader
How frequently have you aided entities in adopting Agile practices	1 Never	2	3	4 occasionally	[✓ 5]	6	7 constantly
Please rate your level of familiarity with general process assessment and/or process improvement	1 None	2	3	[✓ 4] participant	5	6	7 leader

SECTION 2: AGILE LEVELS

[illegible]					
The 5 Agile Levels are comprehensive enough to depict the various states and levels most organizations could be at relative to agility?				✓	
The 5 Agile Levels are defined in a valid and logical order?				✓	
One of our objectives is to make sure that the agile levels are practical and can be used in industry. In light of this, to what extent would you agree that these Agile Levels can be used to classify, and/or aid in the transition of, currently employed software development efforts?			✓		
The classification of agility into five Agile Levels as presented would be beneficial to the software engineering industry?			✓		
Each of the Agile Levels presented encompass a set of agile practices and concepts. To what extent would you agree that those practices and concepts are relevant and correctly assigned to their respective agile levels?			✓		

Would you add, remove or redefine any of the 5 agile levels? If so please explain why.

Do you of any further comments about the Agile Levels?

Promising idea. Too early to tell if this will be beneficial. I don't foresee agreement ever occurring among the Agile thought leaders about the Agile Levels.

OU17

SECTION 3: THE OVERALL PROCESS FRAMEWORK

	strongly disagree	slightly disagree	neither disagree nor agree	slightly agree	strongly agree
[illegible]					
Overall objective of this process framework				✓	
Discontinuing Factors					✓
5 Agile Levels					✓
Project –Level Assessment					✓
Organizational Assessment					✓
Gap Analysis					✓
[illegible]					
One of our objectives is to make sure that the process framework is practical and can be used in industry. In light of this to what extent would you agree that process framework would be practical and feasible to employ?			✓		
[illegible]					
The process framework is beneficial to the software engineering industry?			✓		
[illegible]					
All the necessary components are present in this process framework in order to achieve its overall goal of aiding an organization in the adoption of agile development for its various projects?				✓	
The steps and activities in the process framework are organized in a logical and valid sequence in order to achieve its overall goal of an assisting agile adoption?				✓	
[illegible]					
Discontinuing Factors			✓		
5 Agile Levels			✓		
Project –Level Assessment					✓
Organizational Assessment					✓
Gap Analysis					✓

Would you add, remove or redefine any components of this process framework? If so please explain why.

Do you have any additional comments about the process framework?

I very much like the ideas, especially the differentiation between Project and Organization "principles".

SECTION 5: FURTHER EVALUTION

[illegible] research?	YES	NO	[✓ MAYBE]

0218

VIRGINIA POLYTECHNIC INSTITUTE AND STATE UNIVERSITY

Virginia Tech

Assessment Questionnaire:
"Process for Adoption of Agile Practices in Projects"

Date: | Reference # *(for archiving purposes):*

SECTION 1: ASSESSOR'S INFORMATION

Name (Optional):

Organization / Institute: C

Official Position: Director of Channels | **Years in position :** 1

Email : (

Phone Number :

Please rate your familiarity with Agile software development	1 Not familiar	2	3	4 somewhat familiar	5	6	7 expert
Please rate your highest level of involvement in development efforts that used Agile practices	1 None	2	3	4 participant	5	6	7 leader
How frequently have you aided entities in adopting Agile practices	1 Never	2	3	4 occasionally	5	6	7 constantly
Please rate your level of familiarity with general process assessment and/or process improvement	1 None	2	3	4 participant	5	6	7 leader

SECTION 2: AGILE LEVELS

	strongly disagree	slightly disagree	neither disagree nor agree	slightly agree	strongly agree
COMPREHENSIVENESS					
The 5 Agile Levels are comprehensive enough to depict the various states and levels most organizations could be at relative to agility?					X
The 5 Agile Levels are defined in a valid and logical order?					X
PRACTICALITY					
One of our objectives is to make sure that the agile levels are practical and can be used in industry. In light of this, to what extent would you agree that these Agile Levels can be used to classify, and/or aid in the transition of, currently employed software development efforts?				X	
NECESSITY					
The classification of agility into five Agile Levels as presented would be needed and beneficial to the software engineering industry?				X	
RELEVANCE					
Each of the Agile Levels presented encompass a set of agile practices and concepts. To what extent would you agree that those practices and concepts are relevant and correctly assigned to their respective agile levels?			X		

Would you add, remove or redefine any of the 5 agile levels? If so please explain why.

Do you of any further comments about the Agile Levels?

OV18

SECTION 3: THE OVERALL PROCESS FRAMEWORK

	strongly disagree	slightly disagree	neither disagree nor agree	slightly agree	strongly agree
UNDERSTANDABILITY					
For the topics listed below designate the degree to which you agree that they are understandable					
Overall objective of this process framework					X
Discontinuing Factors				X	
5 Agile Levels					X
Project –Level Assessment					X
Organizational Assessment				X	
Gap Analysis					X
PRACTICALITY					
One of our objectives is to make sure that the process framework is practical and can be used in industry. In light of this to what extent would you agree that process framework would be practical and feasible to employ?				X	
NECESSITY					
The process framework is needed and beneficiary in the software engineering industry?					X
COMPLETENESS					
All the necessary components are present in this process framework in order to achieve its overall goal of aiding an organization in the adoption of agile development for its various projects?				X	
The steps and activities in the process framework are organized in a logical and valid sequence in order to achieve its overall goal of an assisting agile adoption?				X	
EFFECTIVENESS					
To what extent do you agree that each of the components in this process framework is n cessary and sufficient for the framework to achieve its purpose?					
Discontinuing Factors				X	
5 Agile Levels					X
Project –Level Assessment					X
Organizational Assessment					X
Gap Analysis					X

Would you add, remove or redefine any components of this process framework? If so please explain why.

Do you have any additional comments about the process framework?

SECTION 5: FURTHER EVALUTION

Would you like to participate in an extensive, more detailed, evaluation of this research?	**YES**	**[✓ NO]**	**MAYBE**

OV14

VIRGINIA POLYTECHNIC INSTITUTE AND STATE UNIVERSITY

Virginia Tech

Assessment Questionnaire:
"Process for Adoption of Agile Practices in Projects"

Date:	Reference # *(for archiving purposes)*:

SECTION 1: ASSESSOR'S INFORMATION

Name (Optional): [redacted]

Organization / Institute: [redacted]

Official Position: CTO **Years in position :** 4

Email : [redacted] **Phone Number :** [redacted]

Please rate your familiarity with Agile software development	1 Not familiar	2	3	4 somewhat familiar	5	6	[✓ 7 expert]
Please rate your highest level of involvement in development efforts that used Agile practices	1 None	2	3	4 participant	5	6	[✓ 7 leader]
How frequently have you aided entities in adopting Agile practices	1 Never	2	3	4 occasionally	5	6	[✓ 7 constantly]
Please rate your level of familiarity with general process assessment and/or process improvement	1 None	2	3	4 participant	5	[✓ 6]	7 leader

SECTION 2: AGILE LEVELS

The 5 Agile Levels are comprehensive enough to depict the various states and levels most organizations could be at relative to agility?					X
The 5 Agile Levels are defined in a valid and logical order?				X	
One of our objectives is to make sure that the agile levels are practical and can be used in industry. In light of this, to what extent would you agree that these Agile Levels can be used to classify, and/or aid in the transition of, currently employed software development efforts?					X
The classification of agility into five Agile Levels as presented would be beneficial to the software engineering industry?					X
Each of the Agile Levels presented encompass a set of agile practices and concepts. To what extent would you agree that those practices and concepts are relevant and correctly assigned to their respective agile levels?				X	

Would you add, remove or redefine any of the 5 agile levels? If so please explain why.

- Change the label - for each level as described in our meeting (non parallel / progressive)

- ~~Practices~~ may not be the right variables to put on the X axis

Principles - Process - Org

- Tech - Strategy

Do you of any further comments about the Agile Levels?

- Wrong label - Not levels - focus, Stage

Level 1 → Level 5

— Has to be a consistent scale whole per ~~[illegible]~~ column

→

→ - Each one should be a scale or continuum

ec. open → closed

Scale/continuum

SECTION 3: THE OVERALL PROCESS FRAMEWORK

	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]					
Overall objective of this process framework					X
Discontinuing Factors					X
5 Agile Levels					X
Project –Level Assessment					X
Organizational Assessment					X
Gap Analysis					X
[illegible]					
One of our objectives is to make sure that the process framework is practical and can be used in industry. In light of this to what extent would you agree that process framework would be practical and feasible to employ?					X
[illegible]					
The process framework is beneficial to the software engineering industry?				X –	Needs an advocate
[illegible]					
All the necessary components are present in this process framework in order to achieve its overall goal of aiding an organization in the adoption of agile development for its various projects?				X –	Body and few Structure
The steps and activities in the process framework are organized in a logical and valid sequence in order to achieve its overall goal of an assisting agile adoption?					X
[illegible]					
Discontinuing Factors				X	
5 Agile Levels					X
Project –Level Assessment				X	
Organizational Assessment					✓
Gap Analysis				X	

Would you add, remove or redefine any components of this process framework? If so please explain why.

No it seems like a valid whole

– So, add the addition of expected value per level would be good
or observed

Do you have any additional comments about the process framework?

– What is it based on?

– How will save them money?

– How will it remove the failures in the consulting Market

– How will it help protect agile from the coming backlash – "We tried Agile and it failed"

– "What is Agile"

SECTION 5: FURTHER EVALUTION

[illegible]	YES	NO	[✓ MAYBE]

About your thesis:

– Compare + Contrast to other adoption models from OD/Learning Organizations

– It is a valid model, but is it right or ground breaking
applied well in another market

–

VIRGINIA POLYTECHNIC INSTITUTE AND STATE UNIVERSITY

Virginia Tech

Assessment Questionnaire:
"Process for Adoption of Agile Practices in Projects"

OV20

Date: 30 Oct 2006	Reference # *(for archiving purposes)*:

SECTION 1: ASSESSOR'S INFORMATION

Name (Optional): [redacted]

Organization / Institute: R[redacted]

Official Position: Agile Coach [redacted]on : ~ 5

Email : [redacted] [redacted]hone Number : [redacted]

Please rate your familiarity with Agile software development	1 Not familiar	2	3	4 somewhat familiar	5	6	[✓ 7 expert]
Please rate your highest level of involvement in development efforts that used Agile practices	1 None	2	3	4 participant	5	6	[✓ 7 leader]
How frequently have you aided entities in adopting Agile practices	1 Never	2	3	4 occasionally	5	6	[✓ 7 constantly]
Please rate your level of familiarity with general process assessment and/or process improvement	1 None	2	3	4 participant	5	[✓ 6]	7 leader

SECTION 2: AGILE LEVELS

The 5 Agile Levels are comprehensive enough to depict the various states and levels most organizations could be at relative to agility?				✓	
The 5 Agile Levels are defined in a valid and logical order?		✓			
One of our objectives is to make sure that the agile levels are practical and can be used in industry. In light of this, to what extent would you agree that these Agile Levels can be used to classify, and/or aid in the transition of, currently employed software development efforts?				✓	
The classification of agility into five Agile Levels as presented would be beneficial to the software engineering industry?				✓	
Each of the Agile Levels presented encompass a set of agile practices and concepts. To what extent would you agree that those practices and concepts are relevant and correctly assigned to their respective agile levels?		✓			

Would you add, remove or redefine any of the 5 agile levels? If so please explain why.

I don't like calling them levels. I also believe that the vertical & horizontal axes are not orthogonal enough; their characteristics look too similar to help me bound clearly what is differentiated in each box.

Do you of any further comments about the Agile Levels?

I was confused about practices in the matrix versus what were examples provided in the presentation given. And I think practices in the matrix should show more growth from top to bottom and more symmetry from side to side.

OV20

SECTION 3: THE OVERALL PROCESS FRAMEWORK

	[illegible]				
[illegible]					
Overall objective of this process framework					✓
Discontinuing Factors					✓
5 Agile Levels		✓			
Project –Level Assessment					✓
Organizational Assessment				✓	
Gap Analysis			✓		
[illegible]					
One of our objectives is to make sure that the process framework is practical and can be used in industry. In light of this to what extent would you agree that process framework would be practical and feasible to employ?				✓	
[illegible]					
The process framework is beneficial to the software engineering industry?				✓	
[illegible]					
All the necessary components are present in this process framework in order to achieve its overall goal of aiding an organization in the adoption of agile development for its various projects?		✓			
The steps and activities in the process framework are organized in a logical and valid sequence in order to achieve its overall goal of an assisting agile adoption?		✓			
[illegible]					
Discontinuing Factors				✓	
5 Agile Levels		✓			
Project –Level Assessment			✓		
Organizational Assessment			✓		
Gap Analysis			✓		

Would you add, remove or redefine any components of this process framework? If so please explain why.

Consider renaming Org assessment to environmental and also add something about the actual program or technology as well as broadening the term "Developer"

Do you have any additional comments about the process framework?

I like the idea of setting a clear evaluation, map and set of practices in place based on your research. I'd like it to be more crisp (though maybe that was just a problem of our short time frame for you to present a brief overview)

SECTION 5: FURTHER EVALUTION

Would you like to [illegible] research?	YES	NO	[✓ MAYBE]

0021

VIRGINIA POLYTECHNIC INSTITUTE AND STATE UNIVERSITY

Virginia Tech

Assessment Questionnaire:
"Process for Adoption of Agile Practices in Projects"

Date: 10-30-2006	Reference # *(for archiving purposes)*:

SECTION 1: ASSESSOR'S INFORMATION

Name (Optional): [redacted]

Organization / Institute: [redacted]

Official Position: Agile Coach | **Years in position :** 2

Email : [redacted] | **Phone Number :** [redacted]

Please rate your familiarity with Agile software development	1 Not familiar	2	3	4 somewhat familiar	5	6	[✓ 7] expert
Please rate your highest level of involvement in development efforts that used Agile practices	1 None	2	3	4 participant	5	6	[✓ 7] leader
How frequently have you aided entities in adopting Agile practices	1 Never	2	3	4 occasionally	5	6	[✓ 7] constantly
Please rate your level of familiarity with general process assessment and/or process improvement	1 None	2	3	4 participant	5	6	[✓ 7] leader

SECTION 2: AGILE LEVELS

	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
COMPREHENSIVENESS					
The 5 Agile Levels are comprehensive enough to depict the various states and levels most organizations could be at relative to agility?					✓
The 5 Agile Levels are defined in a valid and logical order?					✓
PRACTICALITY					
One of our objectives is to make sure that the agile levels are practical and can be used in industry. In light of this, to what extent would you agree that these Agile Levels can be used to classify, and/or aid in the transition of, currently employed software development efforts?					✓
NECESSITY					
The classification of agility into five Agile Levels as presented would be beneficial to the software engineering industry?				✓	
RELEVANCE					
Each of the Agile Levels presented encompass a set of agile practices and concepts. To what extent would you agree that those practices and concepts are relevant and correctly assigned to their respective agile levels?				✓	

Would you add, remove or redefine any of the 5 agile levels? If so please explain why.

change to Jean's wording, eliminate use of "levels"

Do you of any further comments about the Agile Levels?

1270

SECTION 3: THE OVERALL PROCESS FRAMEWORK

	Strongly Disagree	Slightly Disagree	Neither agree nor disagree	Slightly Agree	Strongly Agree
UNDERSTANDABILITY [illegible]					
Overall objective of this process framework				✓	
Discontinuing Factors					✓
5 Agile Levels					✓
Project –Level Assessment			✓		
Organizational Assessment		✓			
Gap Analysis			✓		
PRACTICALITY					
One of our objectives is to make sure that the process framework is practical and can be used in industry. In light of this to what extent would you agree that process framework would be practical and feasible to employ?					✓
NECESSITY					
The process framework is beneficial to the software engineering industry?				✓	
COMPLETENESS					
All the necessary components are present in this process framework in order to achieve its overall goal of aiding an organization in the adoption of agile development for its various projects?			✓		
The steps and activities in the process framework are organized in a logical and valid sequence in order to achieve its overall goal of an assisting agile adoption?			✓		
EFFECTIVENESS [illegible]					
Discontinuing Factors					✓
5 Agile Levels					✓
Project –Level Assessment				✓	
Organizational Assessment			✓		
Gap Analysis			✓		

Would you add, remove or redefine any components of this process framework? If so please explain why.

____

____

____

____

Do you have any additional comments about the process framework?

____

____

____

____

SECTION 5: FURTHER EVALUTION

Would you like to participate in a more detailed evaluation of this research?	YES	NO	[✓ MAYBE]

VIRGINIA POLYTECHNIC INSTITUTE AND STATE UNIVERSITY

Virginia Tech

Assessment Questionnaire:
"Process for Adoption of Agile Practices in Projects"

OV22

Date: 10/30/06

Reference # *(for archiving purposes)*:

SECTION 1: ASSESSOR'S INFORMATION

Name (Optional): [redacted]

Organization / Institute: [redacted]

Official Position: PRODUCT MANAGER

Years in position : 0 (WAS AGILE COACH)

Email : [redacted]

Phone Numbe [redacted]

Please rate your familiarity with Agile software development	1 Not familiar, 2, 3, 4 somewhat familiar, 5, 6, [✓ 7 expert]
Please rate your highest level of involvement in development efforts that used Agile practices	1 None, 2, 3, 4 participant, 5, 6, [✓ 7 leader]
How frequently have you aided entities in adopting Agile practices	1 Never, 2, 3, 4 occasionally, 5, 6, [✓ 7 constantly]
Please rate your level of familiarity with general process assessment and/or process improvement	1 None, 2, 3, 4 participant, 5, [✓ 6], 7 leader

SECTION 2: AGILE LEVELS

	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
COMPREHENSIVENESS					
The 5 Agile Levels are comprehensive enough to depict the various states and levels most organizations could be at relative to agility?				X	
The 5 Agile Levels are defined in a valid and logical order?				X	
PRACTICALITY					
One of our objectives is to make sure that the agile levels are practical and can be used in industry. In light of this, to what extent would you agree that these Agile Levels can be used to classify, and/or aid in the transition of, currently employed software development efforts?				X	
NECESSITY					
The classification of agility into five Agile Levels as presented would be beneficial to the software engineering industry?		X			
[illegible]					
Each of the Agile Levels presented encompass a set of agile practices and concepts. To what extent would you agree that those practices and concepts are relevant and correctly assigned to their respective agile levels?	X				

Would you add, remove or redefine any of the 5 agile levels? If so please explain why.

Do you of any further comments about the Agile Levels?

0722

SECTION 3: THE OVERALL PROCESS FRAMEWORK

	Strongly [illegible]	Slightly [illegible]	[illegible]	Slightly [illegible]	Strongly [illegible]
UNDERSTANDABILITY [illegible]					
Overall objective of this process framework				X	
Discontinuing Factors				X	
5 Agile Levels				X	
Project –Level Assessment					X
Organizational Assessment					X
Gap Analysis					X
PRACTICALITY					
One of our objectives is to make sure that the process framework is practical and can be used in industry. In light of this to what extent would you agree that process framework would be practical and feasible to employ?				X	
[illegible]					
The process framework is beneficial to the software engineering industry?			X		
COMPLETENESS					
All the necessary components are present in this process framework in order to achieve its overall goal of aiding an organization in the adoption of agile development for its various projects?	X				
The steps and activities in the process framework are organized in a logical and valid sequence in order to achieve its overall goal of an assisting agile adoption?				X	
[illegible]					
Discontinuing Factors				X	
5 Agile Levels				X	
Project –Level Assessment				X	
Organizational Assessment				X	
Gap Analysis				X	

Would you add, remove or redefine any components of this process framework? If so please explain why.

Do you have any additional comments about the process framework?

LARGE RISK THIS WILL BE ABUSED. I WOULD NEED TO SEE EVIDENCE OF RESEARCH INTO OTHER PROCESS FRAMEWORKS INCLUDING A STUDY OF REASONS FOR THEIR FAILURE. TELLING ME ~~#~~ HOW THIS FRAMEWORK IS DIFFERENT FROM CMM IS NOT ENOUGH – I WANT TO KNOW HOW YOU KNOW YOU WILL AVOID THE SAME FAILINGS.

SECTION 5: FURTHER EVALUTION

Would you like to participate in [illegible] research?	[✓ YES]	NO	MAYBE

VIRGINIA POLYTECHNIC INSTITUTE AND STATE UNIVERSITY

Virginia Tech

0023

Assessment Questionnaire:
"Process for Adoption of Agile Practices in Projects"

Date:	Reference # *(for archiving purposes)*:

SECTION 1: ASSESSOR'S INFORMATION

Name (Optional):			
Organization / Institute:			
Official Position: Managing Architect		**Years in position :** 1	
Email		**one Number :**	

Please rate your familiarity with Agile software development	1 Not familiar	2	3	4 somewhat familiar	5	[✓ 6]	7 expert
Please rate your highest level of involvement in development efforts that used Agile practices	1 None	2	3	4 participant	5	6	[✓ 7 leader]
How frequently have you aided entities in adopting Agile practices	1 Never	2	3	4 occasionally	[✓ 5]	6	7 constantly
Please rate your level of familiarity with general process assessment and/or process improvement	1 None	2	3	4 participant	5	6	[✓ 7 leader]

SECTION 2: AGILE LEVELS

	strongly disagree	slightly disagree	neither disagree nor agree	slightly agree	strongly agree
COMPREHENSIVENESS					
The 5 Agile Levels are comprehensive enough to depict the various states and levels most organizations could be at relative to agility?				X	
The 5 Agile Levels are defined in a valid and logical order?				X	
PRACTICALITY					
One of our objectives is to make sure that the agile levels are practical and can be used in industry. In light of this, to what extent would you agree that these Agile Levels can be used to classify, and/or aid in the transition of, currently employed software development efforts?					X
NECESSITY					
The classification of agility into five Agile Levels as presented would be needed and beneficial to the software engineering industry?					X
RELEVANCE					
Each of the Agile Levels presented encompass a set of agile practices and concepts. To what extent would you agree that those practices and concepts are relevant and correctly assigned to their respective agile levels?				X	

Would you add, remove or redefine any of the 5 agile levels? If so please explain why.

Don't tie OO to Agile, they don't depend on each other

Do you of any further comments about the Agile Levels?

The idea of Agile levels is really good.
This is a good start but needs to be tweaked.

OV23

SECTION 3: THE OVERALL PROCESS FRAMEWORK

	strongly disagree	slightly disagree	neither disagree nor agree	slightly agree	strongly agree
UNDERSTANDABILITY					
For the topics listed below designate the degree to which you agree that they are understandable					
Overall objective of this process framework					X
Discontinuing Factors		X			
5 Agile Levels				X	
Project –Level Assessment				X	
Organizational Assessment				X	
Gap Analysis					X
PRACTICALITY					
One of our objectives is to make sure that the process framework is practical and can be used in industry. In light of this to what extent would you agree that process framework would be practical and feasible to employ?					X
NECESSITY					
The process framework is needed and beneficiary in the software engineering industry?					X
COMPLETENESS					
All the necessary components are present in this process framework in order to achieve its overall goal of aiding an organization in the adoption of agile development for its various projects?			X		
The steps and activities in the process framework are organized in a logical and valid sequence in order to achieve its overall goal of an assisting agile adoption?				X	
EFFECTIVENESS					
To what extent do you agree that each of the components in this process framework is necessary and sufficient for the framework to achieve its purpose?					
Discontinuing Factors					X
5 Agile Levels					X
Project –Level Assessment					X
Organizational Assessment					X
Gap Analysis					X

needs work

Would you add, remove or redefine any components of this process framework? If so please explain why.

refine discontinuing factors, maybe add way ~~to~~ influence and change
make specific practices examples rather than determiners

Do you have any additional comments about the process framework?

This is a really good start, it is going along the right road

SECTION 5: FURTHER EVALUTION

Would you like to participate in an extensive, more detailed, evaluation of this research?	[✓ **YES**]	**NO**	**MAYBE**

VIRGINIA POLYTECHNIC INSTITUTE AND STATE UNIVERSITY

Virginia Tech

Assessment Questionnaire:
"Process for Adoption of Agile Practices in Projects"

0024

Date:	Reference # *(for archiving purposes):*

SECTION 1: ASSESSOR'S INFORMATION

Name (Optional):

Organization / Institute: DIGITAL FOCUS

Official Position: SENIOR DEVELOPER **Years in position :** 2

Email : **Phone Number :**

Please rate your familiarity with Agile software development	1 Not familiar	2	3	4 somewhat familiar	5	[✓ 6]	7 expert
Please rate your highest level of involvement in development efforts that used Agile practices	1 None	2	3	4 participant	5	[✓ 6]	7 leader
How frequently have you aided entities in adopting Agile practices	1 Never	2	3	4 occasionally	[✓ 5]	6	7 constantly
Please rate your level of familiarity with general process assessment and/or process improvement	1 None	2	3	4 participant	[✓ 5]	6	7 leader

SECTION 2: AGILE LEVELS

	strongly disagree	slightly disagree	neither disagree nor agree	slightly agree	strongly agree
COMPREHENSIVENESS					
The 5 Agile Levels are comprehensive enough to depict the various states and levels most organizations could be at relative to agility?					✓
The 5 Agile Levels are defined in a valid and logical order?				✓	
PRACTICALITY					
One of our objectives is to make sure that the agile levels are practical and can be used in industry. In light of this, to what extent would you agree that these Agile Levels can be used to classify, and/or aid in the transition of, currently employed software development efforts?					✓
NECESSITY					
The classification of agility into five Agile Levels as presented would be needed and beneficial to the software engineering industry?			✓		
RELEVANCE					
Each of the Agile Levels presented encompass a set of agile practices and concepts. To what extent would you agree that those practices and concepts are relevant and correctly assigned to their respective agile levels?			✓		

Would you add, remove or redefine any of the 5 agile levels? If so please explain why.

Do you of any further comments about the Agile Levels?

OV24

SECTION 3: THE OVERALL PROCESS FRAMEWORK

	strongly disagree	slightly disagree	neither disagree nor agree	slightly agree	strongly agree
UNDERSTANDABILITY					
For the topics listed below designate the degree to which you agree that they are understandable					
Overall objective of this process framework				✓	
Discontinuing Factors				✓	
5 Agile Levels				✓	
Project –Level Assessment			~~✓~~	✓	
Organizational Assessment				✓	
Gap Analysis				✓	
PRACTICALITY					
One of our objectives is to make sure that the process framework is practical and can be used in industry. In light of this to what extent would you agree that process framework would be practical and feasible to employ?				✓	
NECESSITY					
The process framework is needed and beneficiary in the software engineering industry?			✓		
COMPLETENESS					
All the necessary components are present in this process framework in order to achieve its overall goal of aiding an organization in the adoption of agile development for its various projects?				✓	
The steps and activities in the process framework are organized in a logical and valid sequence in order to achieve its overall goal of an assisting agile adoption?				✓	
EFFECTIVENESS					
To what extent do you agree that each of the components in this process framework is necessary and sufficient for the framework to achieve its purpose?					
Discontinuing Factors					✓
5 Agile Levels				✓	
Project –Level Assessment				✓	
Organizational Assessment				✓	
Gap Analysis				✓	

Would you add, remove or redefine any components of this process framework? If so please explain why.

Do you have any additional comments about the process framework?

SECTION 5: FURTHER EVALUTION

Would you like to participate in an extensive, more detailed, evaluation of this research?	**YES**	**NO**	[✓ **MAYBE**]

0V25

VIRGINIA POLYTECHNIC INSTITUTE AND STATE UNIVERSITY

Virginia Tech

Assessment Questionnaire:
"Process for Adoption of Agile Practices in Projects"

Date:	Reference # *(for archiving purposes):*

SECTION 1: ASSESSOR'S INFORMATION

Name (Optional):	
Organization / Institute:	
Official Position: Developer	**Years in position :**
Email	**Phone Number :**

Please rate your familiarity with Agile software development	1 Not familiar	2	3	4 somewhat familiar	[✓ 5]	6	7 expert
Please rate your highest level of involvement in development efforts that used Agile practices	1 None	2	3	4 participant	5	[✓ 6]	7 leader
How frequently have you aided entities in adopting Agile practices	[✓ 1] Never	2	3	4 occasionally	5	6	7 constantly
Please rate your level of familiarity with general process assessment and/or process improvement	1 None	[✓ 2]	3	4 participant	5	6	7 leader

SECTION 2: AGILE LEVELS

	strongly disagree	slightly disagree	neither disagree nor agree	slightly agree	strongly agree
COMPREHENSIVENESS					
The 5 Agile Levels are comprehensive enough to depict the various states and levels most organizations could be at relative to agility?					✓
The 5 Agile Levels are defined in a valid and logical order?					✓
PRACTICALITY					
One of our objectives is to make sure that the agile levels are practical and can be used in industry. In light of this, to what extent would you agree that these Agile Levels can be used to classify, and/or aid in the transition of, currently employed software development efforts?					✓
NECESSITY					
The classification of agility into five Agile Levels as presented would be needed and beneficial to the software engineering industry?					✓
RELEVANCE					
Each of the Agile Levels presented encompass a set of agile practices and concepts. To what extent would you agree that those practices and concepts are relevant and correctly assigned to their respective agile levels?				✓	

Would you add, remove or redefine any of the 5 agile levels? If so please explain why.

Do you of any further comments about the Agile Levels?

UV25

SECTION 3: THE OVERALL PROCESS FRAMEWORK

	strongly disagree	slightly disagree	neither disagree nor agree	slightly agree	strongly agree
UNDERSTANDABILITY					
For the topics listed below designate the degree to which you agree that they are understandable					
Overall objective of this process framework					✓
Discontinuing Factors					
5 Agile Levels					
Project –Level Assessment				✓	
Organizational Assessment		✓			
Gap Analysis			✓		
PRACTICALITY					
One of our objectives is to make sure that the process framework is practical and can be used in industry. In light of this to what extent would you agree that process framework would be practical and feasible to employ?				✓	
NECESSITY					
The process framework is needed and beneficiary in the software engineering industry?					✓
COMPLETENESS					
All the necessary components are present in this process framework in order to achieve its overall goal of aiding an organization in the adoption of agile development for its various projects?				✓	
The steps and activities in the process framework are organized in a logical and valid sequence in order to achieve its overall goal of an assisting agile adoption?					
EFFECTIVENESS					
To what extent do you agree that each of the components in this process framework is necessary and sufficient for the framework to achieve its purpose?					
Discontinuing Factors					✓
5 Agile Levels					✓
Project –Level Assessment					✓
Organizational Assessment					✓
Gap Analysis					✓

Would you add, remove or redefine any components of this process framework? If so please explain why.

Do you have any additional comments about the process framework?

SECTION 5: FURTHER EVALUTION

Would you like to participate in an extensive, more detailed, evaluation of this research?	[✓ **YES**]	**NO**	**MAYBE**

VIRGINIA POLYTECHNIC INSTITUTE AND STATE UNIVERSITY

Virginia Tech

Assessment Questionnaire:
"Process for Adoption of Agile Practices in Projects"

CV26

Date: | Reference # *(for archiving purposes)*:

SECTION 1: ASSESSOR'S INFORMATION

Name (Optional):

Organization / Institute:

Official Position: Principal / Project Mgr | **Years in position :**

Email | **Phone Number :**

Please rate your familiarity with Agile software development	1 Not familiar	2	3	4 somewhat familiar	5	[✓ 6]	7 expert
Please rate your highest level of involvement in development efforts that used Agile practices	1 None	2	3	4 participant	5	6	[✓ 7 leader]
How frequently have you aided entities in adopting Agile practices	1 Never	2	3	4 occasionally	5	[✓ 6]	7 constantly
Please rate your level of familiarity with general process assessment and/or process improvement	1 None	2	3	4 participant	5	[✓ 6]	7 leader

SECTION 2: AGILE LEVELS

	strongly disagree	slightly disagree	neither disagree nor agree	slightly agree	strongly agree
COMPREHENSIVENESS					
The 5 Agile Levels are comprehensive enough to depict the various states and levels most organizations could be at relative to agility?				✓	
The 5 Agile Levels are defined in a valid and logical order?		✓			
PRACTICALITY					
One of our objectives is to make sure that the agile levels are practical and can be used in industry. In light of this, to what extent would you agree that these Agile Levels can be used to classify, and/or aid in the transition of, currently employed software development efforts?				✓	
NECESSITY					
The classification of agility into five Agile Levels as presented would be needed and beneficial to the software engineering industry?		✓			
RELEVANCE					
Each of the Agile Levels presented encompass a set of agile practices and concepts. To what extent would you agree that those practices and concepts are relevant and correctly assigned to their respective agile levels?		✓			

Would you add, remove or redefine any of the 5 agile levels? If so please explain why.

I think that the number of levels is practical. However the precise concept & practice in each level would need to be evaluated to determine what changes are appropriate – thus the slight disagreement

Do you of any further comments about the Agile Levels?

would need to be analyzed in more detail to give an answer

OV26

SECTION 3: THE OVERALL PROCESS FRAMEWORK

	strongly disagree	slightly disagree	neither disagree nor agree	slightly agree	strongly agree
UNDERSTANDABILITY					
For the topics listed below designate the degree to which you agree that they are understandable					
Overall objective of this process framework					✓
Discontinuing Factors				✓	
5 Agile Levels				✓	
Project –Level Assessment					✓
Organizational Assessment					✓
Gap Analysis				✓	
PRACTICALITY					
One of our objectives is to make sure that the process framework is practical and can be used in industry. In light of this to what extent would you agree that process framework would be practical and feasible to employ?				✓	
NECESSITY					
The process framework is needed and beneficiary in the software engineering industry?			✓		
COMPLETENESS					
All the necessary components are present in this process framework in order to achieve its overall goal of aiding an organization in the adoption of agile development for its various projects?		✓			
The steps and activities in the process framework are organized in a logical and valid sequence in order to achieve its overall goal of an assisting agile adoption?					✓
EFFECTIVENESS					
To what extent do you agree that each of the components in this process framework is necessary and sufficient for the framework to achieve its purpose?					
Discontinuing Factors					✓
5 Agile Levels				✓	
Project –Level Assessment				✓	
Organizational Assessment				✓	
Gap Analysis				✓	

Would you add, remove or redefine any components of this process framework? If so please explain why.

The discontinuing factors should not include life/mission critical systems. These are not methodology dependent! Classic waterfall approach is merely an artifactof DoD contracting and not inherently reliable!

Do you have any additional comments about the process framework?

SECTION 5: FURTHER EVALUTION

Would you like to participate in an extensive, more detailed, evaluation of this research?	[✓ YES]	NO	MAYBE

OV2X

VIRGINIA POLYTECHNIC INSTITUTE AND STATE UNIVERSITY

Virginia Tech

Assessment Questionnaire:
"Process for Adoption of Agile Practices in Projects"

Date: 08/03/06	Reference # *(for archiving purposes):*

SECTION 1: ASSESSOR'S INFORMATION

Name (Optional):

Organization / Institute:

Official Position: Software Engineer	**Years in position :** 2
Em:	**Phone Number :**

Please rate your familiarity with Agile software development	1 Not familiar	2	3	4 somewhat familiar	5	[✓ 6]	7 expert
Please rate your highest level of involvement in development efforts that used Agile practices	1 None	2	3	4 participant	5	[✓ 6]	7 leader
How frequently have you aided entities in adopting Agile practices	1 Never	2	3	4 occasionally	[✓ 5]	6	7 constantly
Please rate your level of familiarity with general process assessment and/or process improvement	1 None	2	3	[✓ 4] participant	5	6	7 leader

SECTION 2: AGILE LEVELS

	strongly disagree	slightly disagree	neither disagree nor agree	slightly agree	strongly agree
COMPREHENSIVENESS					
The 5 Agile Levels are comprehensive enough to depict the various states and levels most organizations could be at relative to agility?				✓	
The 5 Agile Levels are defined in a valid and logical order?					✓
PRACTICALITY					
One of our objectives is to make sure that the agile levels are practical and can be used in industry. In light of this, to what extent would you agree that these Agile Levels can be used to classify, and/or aid in the transition of, currently employed software development efforts?		✓			
NECESSITY					
The classification of agility into five Agile Levels as presented would be needed and beneficial to the software engineering industry?		✓			
RELEVANCE					
Each of the Agile Levels presented encompass a set of agile practices and concepts. To what extent would you agree that those practices and concepts are relevant and correctly assigned to their respective agile levels?	✓				

Would you add, remove or redefine any of the 5 agile levels? If so please explain why.

Do you of any further comments about the Agile Levels?

UVZT

SECTION 3: THE OVERALL PROCESS FRAMEWORK

	strongly disagree	slightly disagree	neither disagree nor agree	slightly agree	strongly agree
UNDERSTANDABILITY					
For the topics listed below designate the degree to which you agree that they are understandable					
Overall objective of this process framework				✓	
Discontinuing Factors				✓	
5 Agile Levels				✓	
Project –Level Assessment				✓	
Organizational Assessment				✓	
Gap Analysis				✓	
PRACTICALITY					
One of our objectives is to make sure that the process framework is practical and can be used in industry. In light of this to what extent would you agree that process framework would be practical and feasible to employ?			✓		
NECESSITY					
The process framework is needed and beneficiary in the software engineering industry?					✓
COMPLETENESS					
All the necessary components are present in this process framework in order to achieve its overall goal of aiding an organization in the adoption of agile development for its various projects?			✓		
The steps and activities in the process framework are organized in a logical and valid sequence in order to achieve its overall goal of an assisting agile adoption?			✓		
EFFECTIVENESS					
To what extent do you agree that each of the components in this process framework is necessary and sufficient for the framework to achieve its purpose?					
Discontinuing Factors					✓
5 Agile Levels	✓				
Project –Level Assessment				✓	
Organizational Assessment				✓	
Gap Analysis			✓		

Would you add, remove or redefine any components of this process framework? If so please explain why.

Do you have any additional comments about the process framework?

SECTION 5: FURTHER EVALUTION

Would you like to participate in an extensive, more detailed, evaluation of this research?	**YES**	**NO**	[✓ **MAYBE**]

OV28

VIRGINIA POLYTECHNIC INSTITUTE AND STATE UNIVERSITY

Virginia Tech

Assessment Questionnaire:
"Process for Adoption of Agile Practices in Projects"

Date:	Reference # *(for archiving purposes)*:

SECTION 1: ASSESSOR'S INFORMATION

Name (Optional): [redacted]							
Organization / Instit [redacted]							
Official Position: Principal	**Years in position :** 5						
Email : [redacted]	**Phone Number**						
Please rate your familiarity with Agile software development	1 Not familiar	2	3	4 somewhat familiar	5	6	[✓ 7] expert
Please rate your highest level of involvement in development efforts that used Agile practices	1 None	2	3	4 participant	5	6	[✓ 7] leader
How frequently have you aided entities in adopting Agile practices	1 Never	2	3	4 occasionally	5	6	[✓ 7] constantly
Please rate your level of familiarity with general process assessment and/or process improvement	1 None	2	3	4 participant	[✓ 5]	6	7 leader

SECTION 2: AGILE LEVELS

COMPREHENSIVENESS					
The 5 Agile Levels are comprehensive enough to depict the various states and levels most organizations could be at relative to agility?				✓	
The 5 Agile Levels are defined in a valid and logical order?				✓	
PRACTICALITY					
One of our objectives is to make sure that the agile levels are practical and can be used in industry. In light of this, to what extent would you agree that these Agile Levels can be used to classify, and/or aid in the transition of, currently employed software development efforts?					✓
NECESSITY					
The classification of agility into five Agile Levels as presented would be needed and beneficial to the software engineering industry?				✓	
RELEVANCE					
Each of the Agile Levels presented encompass a set of agile practices and concepts. To what extent would you agree that those practices and concepts are relevant and correctly assigned to their respective agile levels?				✓	

Would you add, remove or redefine any of the 5 agile levels? If so please explain why.

Do you of any further comments about the Agile Levels?

I like the overall indicators & factors. But I see different ones as important in different contexts, so the level thing seems limiting.

There's also an extent to which some practices you have in later levels can be used to drive eg. trust, collaboration, making them have more leverage than would otherwise be apparent.

OV28

SECTION 3: THE OVERALL PROCESS FRAMEWORK

	strongly disagree	slightly disagree	neither disagree nor agree	slightly agree	strongly agree
UNDERSTANDABILITY [illegible]					
Overall objective of this process framework					✓
Discontinuing Factors					✓
5 Agile Levels				✓	
Project –Level Assessment					✓
Organizational Assessment					✓
Gap Analysis					✓
PRACTICALITY					
One of our objectives is to make sure that the process framework is practical and can be used in industry. In light of this to what extent would you agree that process framework would be practical and feasible to employ?					✓
NECESSITY					
The process framework is needed and beneficiary in the software engineering industry?					✓
COMPLETENESS					
All the necessary components are present in this process framework in order to achieve its overall goal of aiding an organization in the adoption of agile development for its various projects?					✓
The steps and activities in the process framework are organized in a logical and valid sequence in order to achieve its overall goal of an assisting agile adoption?					✓
EFFECTIVENESS [illegible]					
Discontinuing Factors				✓	
5 Agile Levels				✓	
Project –Level Assessment					✓
Organizational Assessment					✓
Gap Analysis					✓

Would you add, remove or redefine any components of this process framework? If so please explain why.

I remain skeptical of the "levels" idea...

Do you have any additional comments about the process framework?

I like a lot of the indicators & areas for assessment

SECTION 5: FURTHER EVALUTION

Would you like to participate in an extensive, more detailed, evaluation of this research?	[✓ YES]	NO	MAYBE

Empty 12-Page Survey (Detailed)

VIRGINIA POLYTECHNIC INSTITUTE AND STATE UNIVERSITY

Assessment Questionnaire:
"Process for Adoption of Agile Practices in Projects"

Date:	Reference # *(for archiving purposes):*						
ASSESSOR'S INFORMATION							
Name (Optional):							
Organization / Institute:							
Official Position:	**Years in position :**						
Email :	**Phone Number :**						
Please rate your familiarity with Agile software development	**1** Not familiar	**2**	**3**	**4** somewhat familiar	**5**	**6**	**7** expert
Please rate your highest level of involvement in development efforts that used Agile practices	**1** None	**2**	**3**	**4** participant	**5**	**6**	**7** leader
How frequently have you aided entities in adopting Agile practices	**1** Never	**2**	**3**	**4** occasionally	**5**	**6**	**7** constantly
Please rate your level of familiarity with general process assessment and/or process improvement	**1** None	**2**	**3**	**4** participant	**5**	**6**	**7** leader

1. STAGE 1: DISCONTINUING FACTORS (PAGES 5-9)

	strongly disagree	slightly disagree	neither disagree nor agree	slightly agree	strongly agree
UNDERSTANDABILITY					
For the topics listed below designate the degree to which you agree that they are understandable					
The 4 discontinuing factors					
The assessment table for each of the discontinuing factors					
The sample indicators					
LEVEL OF DETAIL					
To what extent do you agree that the material about discontinuing factors is presented at a sufficient level of detail?					
EFFECTIVENESS					
To what extent do you agree that the 4 discontinuing factors are sufficient to fulfill the objective of identifying all the major showstoppers that might be present before adopting an agile process?					
PRACTICALITY					
One of our objectives is to make sure that the proposed assessment framework and indicators are practical and can be used in industry. In light of this to what extent would you agree that these factors and indicators can be used practically?					
RELEVANCE					
Each of the discontinuing factors is associated with the set of sample questions or indicators. To what extent do you agree that those sample indicators are relevant and valid for the assessment of the discontinuing factors?					

Would you add or remove any factors from the proposed set of discontinuing factors? If so please explain why.

Do you have any further comments about the discontinuing factors presented in this section

2. STAGE 2: PROJECT – LEVEL ASSESSMENT (PAGES 10-16)

	strongly disagree	slightly disagree	neither disagree nor agree	slightly agree	strongly agree
UNDERSTANDABILITY					
For the topics listed below designate the degree to which you agree that they are understandable					
The idea behind assessing project-level agile constraints (project-level assessment)					
The actual constraining agile practices and concepts that are assessed for each of the five agile levels					
The sample indicators and questionnaires used for the assessment of the constraining agile practices and concepts					
LEVEL OF DETAIL					
To what extent do you agree that the material about project – level assessment is presented at a sufficient level of detail?					
COMPREHENSIVENESS					
Project characteristics related to agility can differ from one project to another even within the same organization. To what extent do you agree that the factors identified in the section about project-level assessment are outside the project's or organization's control					
To what extent do you agree that the factors presented in this section sufficiently represent all project characteristics that could constrain the potential level of agility of any project?					
PRACTICALITY					
One of our objectives is to make sure that the project level agile characteristics identified in this section is truly reflective of what can be encountered in industry. In light of this to what extent would you agree that these project level agile characteristics would in real life constrain the level of agility that a project may aspire to?					
To what extent do you agree to the importance of assessing project level agile characteristics in order to determine the highest level of agility a project may hope to adopt?					
RELEVANCE					
Each of the project level agile characteristics presented in this section was associated to one of the five agile levels. To what extent do you agree that the project level agile characteristics were identified from their correct and relevant agile level?					

Would you add or remove any project level agile characteristics? If so please explain why.

____

____

____

____

Do you of any further comments about the project level agile characteristics?

____

____

____

____

3.1 STAGE 3: ORGANIZATIONAL READINESS FOR AGILE LEVEL 1 (PAGES 18-25)

	strongly disagree	slightly disagree	neither disagree nor agree	slightly agree	strongly agree
UNDERSTANDABILITY					
For the topics listed below designate the degree to which you agree that they are understandable					
The objective or theme of Agile Level 1					
The agile practices and concepts identified in Agile Level 1					
The sample indicators and questions related to each agile practice or concept					
LEVEL OF DETAIL					
To what extent do you agree that the material about Agile Level 1 is presented at a sufficient level of detail?					
PRACTICALITY					
One of our objectives is to make sure that the Agile Levels are practical and can be used in industry. In light of this to what extent would you agree that if the agile practices and concepts presented in agile level 1 were used during the project that they would enhance the overall communication and collaboration?					
To what extent do you agree that the results of the questionnaires (indicators) legitimately reflect the readiness of the organization to adopt the agile practices and concepts identified in agile level 1?					
EFFECTIVENESS					
To what extent do you agree that the agile practices and concepts identified in Agile Level 1 are sufficient enough to achieve the objective of this agile level which is enhancing communication and collaboration?					
To what extent would you agree that the first level of agility should focus on enhancing communication and collaboration?					
RELEVANCE					
To what extent do you agree that the agile practices in Agile Level 1 are relevant to their associated agile principals?					
Collaborative planning					
Collaborative teams					
Empowered and motivated teams					
Coding standards					
Knowledge-Sharing tools					
Task volunteering not task assignment					
For each of the agile practices and concept in this level, to what extent do you agree that the sample indicators and questions are relevant and correctly linked to their respective agile practice or concept?					
Collaborative planning					
Collaborative teams					
Empowered and motivated teams					
Coding standards					
Knowledge-Sharing tools					
Task volunteering not task assignment					

Would you add or remove any agile practices or concepts to this agile level? If so please explain why.

____

____

____

____

Do you of any further comments about agile level 1

____

____

____

____

3.2 STAGE 3: ORGANIZATIONAL READINESS FOR AGILE LEVEL 2 (PAGES 26-34)

	strongly disagree	slightly disagree	neither disagree nor agree	slightly agree	strongly agree
UNDERSTANDABILITY					
For the topics listed below designate the degree to which you agree that they are understandable					
The objective or theme of Agile Level 2					
The agile practices and concepts identified in Agile Level 2					
The sample indicators and questions related to each agile practice or concept					
LEVEL OF DETAIL					
To what extent do you agree that the material about Agile Level 2 is presented at a sufficient level of detail?					
PRACTICALITY					
One of our objectives is to make sure that the Agile Levels are practical and can be used in industry. In light of this to what extent would you agree that if the agile practices and concepts presented in agile level 2 were used during the project that they would better ensure delivering software early and continuously?					
To what extent do you agree that the results of the questionnaires (indicators) legitimately reflect the readiness of the organization to adopt the agile practices and concepts identified in agile level 2?					
EFFECTIVENESS					
To what extent do you agree that the agile practices and concepts identified in Agile Level 1 are sufficient enough to achieve the objective of this agile level which is delivering software early and continuously?					
To what extent would you agree that the first level of agility should focus on delivering software early and continuously?					
RELEVANCE					
To what extent do you agree that the agile practices in Agile Level 2 are relevant to their associated agile principals?					
Evolutionary requirements					
Continuous delivery (incremental- iterative development)					
Planning get different levels					
Software configuration management					
Tracking the iteration progress through working software					
No Big Design Up Front					
For each of the agile practices and concept in this level, to what extent do you agree that the sample indicators and questions are relevant and correctly linked to their respective agile practice or concept?					
Evolutionary requirements					
Continuous delivery (incremental- iterative development)					
Planning get different levels					
Software configuration management					
Tracking the iteration progress through working software					
No Big Design Up Front					

Would you add or remove any agile practices or concepts to this agile level? If so please explain why.

____

____

____

____

Do you of any further comments about agile level 2

____

____

____

____

3.3 STAGE 3: ORGANIZATIONAL READINESS FOR AGILE LEVEL 3 (PAGES 35-41)

	strongly disagree	slightly disagree	neither disagree nor agree	slightly agree	strongly agree
UNDERSTANDABILITY					
For the topics listed below designate the degree to which you agree that they are understandable					
The objective or theme of Agile Level 3					
The agile practices and concepts identified in Agile Level 3					
The sample indicators and questions related to each agile practice or concept					
LEVEL OF DETAIL					
To what extent do you agree that the material about Agile Level 3 is presented at a sufficient level of detail?					
PRACTICALITY					
One of our objectives is to make sure that the Agile Levels are practical and can be used in industry. In light of this to what extent would you agree that if the agile practices and concepts presented in agile level 3 were used during the project that they would aid in the production of quality working software?					
To what extent do you agree that the results of the questionnaires (indicators) legitimately reflect the readiness of the organization to adopt the agile practices and concepts identified in agile level 3?					
EFFECTIVENESS					
To what extent do you agree that the agile practices and concepts identified in Agile Level 3 are sufficient enough to achieve the objective of this agile level which is producing quality software?					
To what extent would you agree that the first level of agility should focus on producing quality software?					
RELEVANCE					
To what extent do you agree that the agile practices in Agile Level 3 are relevant to their associated agile principals?					
Risk driven iterations					
Continuous improvement					
Self-organizing teams					
The use of true object oriented design and construction					
Continuous integration					
Maintaining the list of all remaining features					
Unit tests					
For each of the agile practices and concept in this level, to what extent do you agree that the sample indicators and questions are relevant and correctly linked to their respective agile practice or concept?					
Risk driven iterations					
Continuous improvement					
Self-organizing teams					
The use of true object oriented design and construction					
Continuous integration					
Maintaining the list of all remaining features					
Unit tests					

Would you add or remove any agile practices or concepts to this agile level? If so please explain why.

Do you of any further comments about agile level 3

3.4 STAGE 3: ORGANIZATIONAL READINESS FOR AGILE LEVEL 4 (PAGES 42-48)

	strongly disagree	slightly disagree	neither disagree nor agree	slightly agree	strongly agree
UNDERSTANDABILITY					
For the topics listed below designate the degree to which you agree that they are understandable					
The objective or theme of Agile Level 4					
The agile practices and concepts identified in Agile Level 4					
The sample indicators and questions related to each agile practice or concept					
LEVEL OF DETAIL					
To what extent do you agree that the material about Agile Level 4 is presented at a sufficient level of detail?					
PRACTICALITY					
One of our objectives is to make sure that the Agile Levels are practical and can be used in industry. In light of this to what extent would you agree that if the agile practices and concepts presented in agile level 4 were used during the project that they would become more responsive to change?					
To what extent do you agree that the results of the questionnaires (indicators) legitimately reflect the readiness of the organization to adopt the agile practices and concepts identified in agile level 4?					
EFFECTIVENESS					
To what extent do you agree that the agile practices and concepts identified in Agile Level 1 are sufficient enough to achieve the objective of this agile level which is responding to change through multiple levels of feedback?					
To what extent would you agree that the first level of agility should focus on responding to change through multiple levels of feedback?					
RELEVANCE					
To what extent do you agree that the agile practices in Agile Level 4 are relevant to their associated agile principals?					
Client driven iterations					
Continuous customer satisfaction feedback					
Reflect and tune process					
Smaller and more frequent releases					
Adaptive planning					
Daily progress tracking meetings					
Agile documentation (from agile modeling)					
User stories					
For each of the agile practices and concept in this level, to what extent do you agree that the sample indicators and questions are relevant and correctly linked to their respective agile practice or concept?					
Client driven iterations					
Continuous customer satisfaction feedback					
Reflect and tune process					
Smaller and more frequent releases					
Adaptive planning					
Daily progress tracking meetings					
Agile documentation (from agile modeling)					
User stories					

Would you add or remove any agile practices or concepts to this agile level? If so please explain why.

____

____

____

Do you of any further comments about agile level 4

____

____

____

3.5 STAGE 3: ORGANIZATIONAL READINESS FOR AGILE LEVEL 5 (PAGES 49-55)

	strongly disagree	slightly disagree	neither disagree nor agree	slightly agree	strongly agree
UNDERSTANDABILITY					
For the topics listed below designate the degree to which you agree that they are understandable					
The objective or theme of Agile Level 5					
The agile practices and concepts identified in Agile Level 5					
The sample indicators and questions related to each agile practice or concept					
LEVEL OF DETAIL					
To what extent do you agree that the material about Agile Level 5 is presented at a sufficient level of detail?					
PRACTICALITY					
One of our objectives is to make sure that the Agile Levels are practical and can be used in industry. In light of this to what extent would you agree that if the agile practices and concepts presented in agile level 5 were used during the project that they would establish a true Agile development environment?					
To what extent do you agree that the results of the questionnaires (indicators) legitimately reflect the readiness of the organization to adopt the agile practices and concepts identified in agile level 5?					
EFFECTIVENESS					
To what extent do you agree that the agile practices and concepts identified in Agile Level 1 are sufficient enough to achieve the objective of this agile level which is establishing a true agile development environment?					
To what extent would you agree that the first level of agility should focus on establishing a true agile development environment?					
RELEVANCE					
To what extent do you agree that the agile practices in Agile Level 5 are relevant to their associated agile principals?					
Low process ceremony					
Agile project estimation					
Paired programming					
Test-driven development					
For each of the agile practices and concept in this level, to what extent do you agree that the sample indicators and questions are relevant and correctly linked to their respective agile practice or concept?					
Low process ceremony					
Agile project estimation					
Paired programming					
Test-driven development					

Would you add or remove any agile practices or concepts to this agile level? If so please explain why.

Do you of any further comments about agile level 5

APPENDIX B: INDICATOR AGGREGATION (PAGES 60-64)

	strongly disagree	slightly disagree	neither disagree nor agree	slightly agree	strongly agree
UNDERSTANDABILITY					
For the topics listed below designate the degree to which you agree that they are understandable					
The directions on how to use the automated method					
How to compute a weight for each indicator					
How to compute the weighed interval					
How to calculate optimistic and pessimistic results					
How to translate the results into a nominal score					
LEVEL OF DETAIL					
To what extent do you agree that the material about the indicator aggregation was presented with a sufficient level of detail?					
PRACTICALITY					
One of our objectives is to make sure that our proposed method for indicator aggregation is practical and can be used in industry. In light of this to what extent would you agree that this approach to indicator aggregation can be used practically?					
RELEVANCE					
If this approach is used to aggregate a set of indicators, to what extent would you agree that the results would legitimately reflect a valid and relevant outcome?					
EFFECTIVENESS					
To what extent do you agree that this approach to indicator aggregation is a sufficient and valid one in aggregating the various sets of indicators throughout the process framework?					

Do you of any further comments about the method of indicator aggregation?

Completed 12-page Surveys (Detailed)

VIRGINIA POLYTECHNIC INSTITUTE AND STATE UNIVERSITY

Virginia Tech

Assessment Questionnaire:
"Process for Adoption of Agile Practices in Projects"

1020

Date: 12-1-6	Reference # *(for archiving purposes)*:							
ASSESSOR'S INFORMATION								
Name (Optional):								
Organization / Institute: Independent consultant								
Official Position: Principal	**Years in position :** 5							
Email :	**Phone Number :**							
Please rate your familiarity with Agile software development	1 Not familiar	2	3	4 somewhat familiar	5	6	[✓ 7] expert
Please rate your highest level of involvement in development efforts that used Agile practices	1 None	2	3	4 participant	5	6	[✓ 7] leader
How frequently have you aided entities in adopting Agile practices	1 Never	2	3	4 occasionally	5	6	[✓ 7] constantly
Please rate your level of familiarity with general process assessment and/or process improvement	1 None	2	3	4 participant	[✓ 5]	6	7 leader

UCC1

1. STAGE 1: DISCONTINUING FACTORS (PAGES 5-9)

	strongly disagree	slightly disagree	neither disagree nor agree	slightly agree	strongly agree
UNDERSTANDABILITY					
For the topics listed below designate the degree to which you agree that they are understandable					
The 4 discontinuing factors				✓	
The assessment table for each of the discontinuing factors					✓
The sample indicators					✓
LEVEL OF DETAIL					
To what extent do you agree that the material about discontinuing factors is presented at a sufficient level of detail?				✓	
EFFECTIVENESS					
To what extent do you agree that the 4 discontinuing factors are sufficient to fulfill the objective of identifying all the major showstoppers that might be present before adopting an agile process?				✓	
PRACTICALITY					
One of our objectives is to make sure that the proposed assessment framework and indicators are practical and can be used in industry. In light of this to what extent would you agree that these factors and indicators can be used practically?					✓
RELEVANCE					
Each of the discontinuing factors is associated with the set of sample questions or indicators. To what extent do you agree that those sample indicators are relevant and valid for the assessment of the discontinuing factors?					✓

Would you add or remove any factors from the proposed set of discontinuing factors? If so please explain why.

Depending on how it's defined, I'm not clear that
mission + life-critical are barriers

Do you have any further comments about the discontinuing factors presented in this section

Are on-time + on-budget supporting or not?
lack of (I would say agile would help w/ these problems)

UE01

2. STAGE 2: PROJECT – LEVEL ASSESSMENT (PAGES 10-16)

	strongly disagree	slightly disagree	neither disagree nor agree	slightly agree	strongly agree
UNDERSTANDABILITY					
For the topics listed below designate the degree to which you agree that they are understandable					
The Idea behind assessing project-level agile constraints (project-level assessment)					✓
The actual constraining agile practices and concepts that are assessed for each of the five agile levels					✓
The sample indicators and questionnaires used for the assessment of the constraining agile practices and concepts					✓
LEVEL OF DETAIL					
To what extent do you agree that the material about project – level assessment is presented at a sufficient level of detail?					✓
COMPREHENSIVENESS					
Project characteristics related to agility can differ from one project to another even within the same organization. To what extent do you agree that the factors identified in the section about project-level assessment are outside the project's or organization's control				✓	
To what extent do you agree that the factors presented in this section sufficiently represent all project characteristics that could constrain the potential level of agility of any project?				✓	
PRACTICALITY					
One of our objectives is to make sure that the project level agile characteristics identified in this section is truly reflective of what can be encountered in industry. In light of this to what extent would you agree that these project level agile characteristics would in real life constrain the level of agility that a project may aspire to?				✓	
To what extent do you agree to the importance of assessing project level agile characteristics in order to determine the highest level of agility a project may hope to adopt?				✓	
RELEVANCE					
Each of the project level agile characteristics presented in this section was associated to one of the five agile levels. To what extent do you agree that the project level agile characteristics were identified from their correct and relevant agile level?	✓				

Would you add or remove any project level agile characteristics? If so please explain why.

Because I don't believe they're pre-requisites to each other!

Do you of any further comments about the project level agile characteristics?

I'm not convinced by the notion of levels

One aspect is that the framework ignores the extent to which some of the practices you've put in advanced levels can create the conditions that make other ones possible. For ex., a team committed to continuous integration & test-driven development may demonstrate progress in a way that makes a customer more willing to "ok" a collaborative contract.

Another challenge is that what's in an upper level for a team may depend on the context &

DT-01

3.1 STAGE 3: ORGANIZATIONAL READINESS FOR AGILE LEVEL 1 (PAGES 18-25)

	strongly disagree	slightly disagree	neither disagree nor agree	slightly agree	strongly agree
UNDERSTANDABILITY					
For the topics listed below designate the degree to which you agree that they are understandable					
The objective or theme of Agile Level 1					✓
The agile practices and concepts identified in Agile Level 1					✓
The sample indicators and questions related to each agile practice or concept					✓
LEVEL OF DETAIL					
To what extent do you agree that the material about Agile Level 1 is presented at a sufficient level of detail?					✓
PRACTICALITY					
One of our objectives is to make sure that the Agile Levels are practical and can be used in industry. In light of this to what extent would you agree that if the agile practices and concepts presented in agile level 1 were used during the project that they would enhance the overall communication and collaboration?					✓
To what extent do you agree that the results of the questionnaires (indicators) legitimately reflect the readiness of the organization to adopt the agile practices and concepts identified in agile level 1?					✓
EFFECTIVENESS					
To what extent do you agree that the agile practices and concepts identified in Agile Level 1 are sufficient enough to achieve the objective of this agile level which is enhancing communication and collaboration?					✓
To what extent would you agree that the first level of agility should focus on enhancing communication and collaboration?					✓
RELEVANCE					
To what extent do you agree that the agile practices in Agile Level 1 are relevant to their associated agile principals?					
Collaborative planning					✓
Collaborative teams					✓
Empowered and motivated teams					✓
Coding standards				✓	
Knowledge-Sharing tools				✓	
Task volunteering not task assignment					✓
For each of the agile practices and concept in this level, to what extent do you agree that the sample indicators and questions are relevant and correctly linked to their respective agile practice or concept?					
Collaborative planning					✓
Collaborative teams					✓
Empowered and motivated teams					✓
Coding standards				✓	
Knowledge-Sharing tools					✓
Task volunteering not task assignment					✓

Would you add or remove any agile practices or concepts to this agile level? If so please explain why.

I don't see "coding std" as that important up-front; I'm OK w/ emergence.
I don't see wiki or blog as important at all. (I like it phrased better in ORI M21)
Assesses People vs. Mgmt & Developers but also need to consider Customer / Product Owner

Do you of any further comments about agile level 1

ORI-D9 seems phrased backwards. (ORI-M11)
ORI-M16 too.

DE01

3.2 STAGE 3: ORGANIZATIONAL READINESS FOR AGILE LEVEL 2 (PAGES 26-34)

	strongly disagree	slightly disagree	neither disagree nor agree	slightly agree	strongly agree
UNDERSTANDABILITY					
For the topics listed below designate the degree to which you agree that they are understandable					
The objective or theme of Agile Level 2					✓
The agile practices and concepts identified in Agile Level 2					✓
The sample indicators and questions related to each agile practice or concept					✓
LEVEL OF DETAIL					
To what extent do you agree that the material about Agile Level 2 is presented at a sufficient level of detail?					✓
PRACTICALITY					
One of our objectives is to make sure that the Agile Levels are practical and can be used in industry. In light of this to what extent would you agree that if the agile practices and concepts presented in agile level 2 were used during the project that they would better ensure delivering software early and continuously?					✓
To what extent do you agree that the results of the questionnaires (indicators) legitimately reflect the readiness of the organization to adopt the agile practices and concepts identified in agile level 2?				✓	
EFFECTIVENESS					
2? To what extent do you agree that the agile practices and concepts identified in Agile Level 1 are sufficient enough to achieve the objective of this agile level which is delivering software early and continuously?					✓
To what extent would you agree that the first level of agility should focus on delivering software early and continuously? — yes, first!					✓
RELEVANCE					
To what extent do you agree that the agile practices in Agile Level 2 are relevant to their associated agile principals?					
Evolutionary requirements					✓
Continuous delivery (incremental- iterative development)					✓
Planning get different levels					✓
Software configuration management				✓	
Tracking the iteration progress through working software					✓
No Big Design Up Front				✓	
For each of the agile practices and concept in this level, to what extent do you agree that the sample indicators and questions are relevant and correctly linked to their respective agile practice or concept?					
Evolutionary requirements					✓
Continuous delivery (incremental- iterative development)					✓
Planning get different levels					✓
Software configuration management					✓
Tracking the iteration progress through working software					✓
No Big Design Up Front				✓	

Would you add or remove any agile practices or concepts to this agile level? If so please explain why.

Config. Mgmt seems more like a basic competency to me.
I'm not clear why a defined dev. process is important; I think agile methods have a
lot to offer very ad-hoc org's.

Do you of any further comments about agile level 2

On Cont. Delivery / People / Mgmt / Stress, I think you want "overseeing" not "overlooking"
Might re-organize: "competence" then "buy-in" then "stress"
I'm not sure all agile leaders would agree with you on "BDUF" I'm OK though :)
Need more definition around "big upfront analysis" is a month big?

DF01

3.3 STAGE 3: ORGANIZATIONAL READINESS FOR AGILE LEVEL 3 (PAGES 35-41)

	strongly disagree	slightly disagree	neither disagree nor agree	slightly agree	strongly agree
UNDERSTANDABILITY					
For the topics listed below designate the degree to which you agree that they are understandable					
The objective or theme of Agile Level 3					✓
The agile practices and concepts identified in Agile Level 3					✓
The sample indicators and questions related to each agile practice or concept					✓
LEVEL OF DETAIL					
To what extent do you agree that the material about Agile Level 3 is presented at a sufficient level of detail?					✓
PRACTICALITY					
One of our objectives is to make sure that the Agile Levels are practical and can be used in industry. In light of this to what extent would you agree that if the agile practices and concepts presented in agile level 3 were used during the project that they would aid in the production of quality working software?					✓
To what extent do you agree that the results of the questionnaires (indicators) legitimately reflect the readiness of the organization to adopt the agile practices and concepts identified in agile level 3?				✓	
EFFECTIVENESS					
To what extent do you agree that the agile practices and concepts identified in Agile Level 3 are sufficient enough to achieve the objective of this agile level which is producing quality software?					✓
To what extent would you agree that the first level of agility should focus on producing quality software? yes, first					✓
RELEVANCE					
To what extent do you agree that the agile practices in Agile Level 3 are relevant to their associated agile principals?					
Risk driven iterations					
Continuous improvement					✓
Self-organizing teams					✓
The use of true object oriented design and construction			✓		
Continuous integration					✓
Maintaining the list of all remaining features				✓	
Unit tests					✓
For each of the agile practices and concept in this level, to what extent do you agree that the sample indicators and questions are relevant and correctly linked to their respective agile practice or concept?					
Risk driven iterations				✓	
Continuous Improvement				✓	
Self-organizing teams					✓
The use of true object oriented design and construction				✓	
Continuous integration					✓
Maintaining the list of all remaining features				✓	
Unit tests					✓

Would you add or remove any agile practices or concepts to this agile level? If so please explain why.

Not sure mgmt's risk attitude is the key – also need it in customer / PO.
Unit testing & continuous integration are very early targets for me – they're in the team's span of control

Do you of any further comments about agile level 3

* Continuous improvement is not just of code – it's also of the process itself.
Existence of backlog is not a requirement by me. I don't object where it's needed or used, but I don't think it's the heart of things.
OO & agile often go together, but I don't think they have to.
→ I completely disagree with OR-A2.

3.4 STAGE 3: ORGANIZATIONAL READINESS FOR AGILE LEVEL 4 (PAGES 42-48)

LEV1

	strongly disagree	slightly disagree	neither disagree nor agree	slightly agree	strongly agree
UNDERSTANDABILITY					
For the topics listed below designate the degree to which you agree that they are understandable					
The objective or theme of Agile Level 4					✓
The agile practices and concepts identified in Agile Level 4					✓
The sample indicators and questions related to each agile practice or concept					✓
LEVEL OF DETAIL					
To what extent do you agree that the material about Agile Level 4 is presented at a sufficient level of detail?					✓
PRACTICALITY					
One of our objectives is to make sure that the Agile Levels are practical and can be used in industry. In light of this to what extent would you agree that if the agile practices and concepts presented in agile level 4 were used during the project that they would become more responsive to change?					✓
To what extent do you agree that the results of the questionnaires (indicators) legitimately reflect the readiness of the organization to adopt the agile practices and concepts identified in agile level 4?					✓
EFFECTIVENESS					
?. To what extent do you agree that the agile practices and concepts identified in Agile Level 1 are sufficient enough to achieve the objective of this agile level which is responding to change through multiple levels of feedback?					✓
To what extent would you agree that the first level of agility should focus on responding to change through multiple levels of feedback?					✓
RELEVANCE					
To what extent do you agree that the agile practices in Agile Level 4 are relevant to their associated agile principals?					
Client driven iterations					✓
Continuous customer satisfaction feedback					✓
Reflect and tune process					✓
Smaller and more frequent releases					✓
Adaptive planning					✓
Daily progress tracking meetings					✓
Agile documentation (from agile modeling)				✓	
User stories				✓	
For each of the agile practices and concept in this level, to what extent do you agree that the sample indicators and questions are relevant and correctly linked to their respective agile practice or concept?					
Client driven iterations					✓
Continuous customer satisfaction feedback					✓
Reflect and tune process					✓
Smaller and more frequent releases					✓
Adaptive planning					✓
Daily progress tracking meetings					✓
Agile documentation (from agile modeling)				✓	
User stories					✓

Would you add or remove any agile practices or concepts to this agile level? If so please explain why.

"Agile modelling" is not accepted by all agile folks; I'd call it one person's take on things.
I don't follow OR4-D6
Rephrase OR4-D7

Do you of any further comments about agile level 4

I disagree with OL4-D14. I'd consider user stories a component of agile
used by XP & often Scrum. (but think of Ron Jeffries' "CCC") Not
all agile methods take this approach.
I'm always a little nervous about "customer" as a word e.g. in 3.4.1.2. ✓

LE 01

3.5 STAGE 3: ORGANIZATIONAL READINESS FOR AGILE LEVEL 5 (PAGES 49-55)

	strongly disagree	slightly disagree	neither disagree nor agree	slightly agree	strongly agree
UNDERSTANDABILITY					
For the topics listed below designate the degree to which you agree that they are understandable					
The objective or theme of Agile Level 5					✓
The agile practices and concepts identified in Agile Level 5					✓
The sample indicators and questions related to each agile practice or concept					✓
LEVEL OF DETAIL					
To what extent do you agree that the material about Agile Level 5 is presented at a sufficient level of detail?					✓
PRACTICALITY					
One of our objectives is to make sure that the Agile Levels are practical and can be used in industry. In light of this to what extent would you agree that if the agile practices and concepts presented in agile level 5 were used during the project that they would establish a true Agile development environment?				✓	
To what extent do you agree that the results of the questionnaires (indicators) legitimately reflect the readiness of the organization to adopt the agile practices and concepts identified in agile level 5?				✓	
EFFECTIVENESS					
To what extent do you agree that the agile practices and concepts identified in Agile Level 1 are sufficient enough to achieve the objective of this agile level which is establishing a true agile development environment?			✓		
To what extent would you agree that the first level of agility should focus on establishing a true agile development environment?					✓
RELEVANCE					
To what extent do you agree that the agile practices in Agile Level 5 are relevant to their associated agile principals?					
Low process ceremony				✓	
\gile project estimation					✓
Paired programming				✓	
Test-driven development					✓
For each of the agile practices and concept in this level, to what extent do you agree that the sample indicators and questions are relevant and correctly linked to their respective agile practice or concept?					
Low process ceremony					✓
Agile project estimation					✓
Paired programming					✓
Test-driven development					✓

Would you add or remove any agile practices or concepts to this agile level? If so please explain why.

I don't buy Estimation / Process / Estimation / Method. I don't see what Estimation / People / Mgmt / Competency matters, if team is doing estimates.
I'm a proponent of pair programming, but ~~not~~ only XP really pushes it.
I push TDD very early.

Do you of any further comments about agile level 5

I don't get OR5-D1 & OR5-D2. I don't see the purpose of reviews as diffusing responsibility.
OR5-D13 is not realistic
OR5-D14 & OR5-D15 aren't critical to me.
OR5-D17 is unlikely to be agreed to even by TDD proponents.
OR5-M8 isn't important to many agile methods
~~[illegible]~~ OR5-M10 seems like it should get "never"; in many agile methods, estimation is a team process.
OR5-M16 needs rework.

APPENDIX B: INDICATOR AGGREGATION (PAGES 60-64)

DE01

	strongly disagree	slightly disagree	neither disagree nor agree	slightly agree	strongly agree
UNDERSTANDABILITY					
For the topics listed below designate the degree to which you agree that they are understandable					
The directions on how to use the automated method				✓	
How to compute a weight for each indicator					✓
How to compute the weighed interval				✓	
How to calculate optimistic and pessimistic results					✓
How to translate the results into a nominal score				✓	
LEVEL OF DETAIL					
To what extent do you agree that the material about the indicator aggregation was presented with a sufficient level of detail?				✓	
PRACTICALITY					
One of our objectives is to make sure that our proposed method for indicator aggregation is practical and can be used in industry. In light of this to what extent would you agree that this approach to indicator aggregation can be used practically?				✓	
RELEVANCE					
If this approach is used to aggregate a set of indicators, to what extent would you agree that the results would legitimately reflect a valid and relevant outcome?				✓	
EFFECTIVENESS					
To what extent do you agree that this approach to indicator aggregation is a sufficient and valid one in aggregating the various sets of indicators throughout the process framework?				✓	

Do you of any further comments about the method of indicator aggregation?

Not clear that "equal weights" is appropriate.
Maybe I'm missing something, but it seems like the evaluation sheet can

VIRGINIA POLYTECHNIC INSTITUTE AND STATE UNIVERSITY

Virginia Tech

Assessment Questionnaire:
"Process for Adoption of Agile Practices in Projects"

[illegible]

Date: 10·28-06

Reference # *(for archiving purposes)*:

ASSESSOR'S INFORMATION

Name (Optional):

Organization / Institute:

Official Position: President

Years in position : 13

Email :

Phone Number :

Please rate your familiarity with Agile software development	1 Not familiar	2	3	4 somewhat familiar	5	6	[✓ 7 expert]
Please rate your highest level of involvement in development efforts that used Agile practices	1 None	2	3	4 participant	5	6	[✓ 7 leader]
How frequently have you aided entities in adopting Agile practices	1 Never	2	3	4 occasionally	5	6	[✓ 7 constantly]
Please rate your level of familiarity with general process assessment and/or process improvement	1 None	2	3	4 participant	5	[✓ 6]	7 leader

DE02

1. STAGE 1: DISCONTINUING FACTORS (PAGES 5-9)

	strongly disagree	slightly disagree	neither disagree nor agree	slightly agree	strongly agree
UNDERSTANDABILITY					
For the topics listed below designate the degree to which you agree that they are understandable					
The 4 discontinuing factors					✓
The assessment table for each of the discontinuing factors					✓
The sample indicators			✓		
LEVEL OF DETAIL					
To what extent do you agree that the material about discontinuing factors is presented at a sufficient level of detail?			✓		
EFFECTIVENESS					
To what extent do you agree that the 4 discontinuing factors are sufficient to fulfill the objective of identifying all the major showstoppers that might be present before adopting an agile process?		~~✓~~		✓	
PRACTICALITY					
One of our objectives is to make sure that the proposed assessment framework and indicators are practical and can be used in industry. In light of this to what extent would you agree that these factors and indicators can be used practically?			✓		
RELEVANCE					
Each of the discontinuing factors is associated with the set of sample questions or indicators. To what extent do you agree that those sample indicators are relevant and valid for the assessment of the discontinuing factors?		✓			

Would you add or remove any factors from the proposed set of discontinuing factors? If so please explain why.

Do you have any further comments about the discontinuing factors presented in this section

some indicators are phrased in the negative – e.g., DC-M4&1 – they'd be more clear without the "not"

the indicators are sometimes too simplistic – e.g., DC-A1. We go through the framework & trace it to the assessor's opinion of whether they are late. there s/b questions about perhaps being on-time but w/ too greatly reduced functionality

2. STAGE 2: PROJECT – LEVEL ASSESSMENT (PAGES 10-16)

	strongly disagree	slightly disagree	neither disagree nor agree	slightly agree	strongly agree
UNDERSTANDABILITY					
For the topics listed below designate the degree to which you agree that they are understandable					
The idea behind assessing project-level agile constraints (project-level assessment)					✓
The actual constraining agile practices and concepts that are assessed for each of the five agile levels					✓
The sample indicators and questionnaires used for the assessment of the constraining agile practices and concepts					✓
LEVEL OF DETAIL					
To what extent do you agree that the material about project – level assessment is presented at a sufficient level of detail?					✓
COMPREHENSIVENESS					
Project characteristics related to agility can differ from one project to another even within the same organization. To what extent do you agree that the factors identified in the section about project-level assessment are outside the project's or organization's control					✓
To what extent do you agree that the factors presented in this section sufficiently represent all project characteristics that could constrain the potential level of agility of any project?				✓	
PRACTICALITY					
One of our objectives is to make sure that the project level agile characteristics identified in this section is truly reflective of what can be encountered in industry. In light of this to what extent would you agree that these project level agile characteristics would in real life constrain the level of agility that a project may aspire to?				✓	
To what extent do you agree to the importance of assessing project level agile characteristics in order to determine the highest level of agility a project may hope to adopt?		✓			
RELEVANCE					
Each of the project level agile characteristics presented in this section was associated to one of the five agile levels. To what extent do you agree that the project level agile characteristics were identified from their correct and relevant agile level?			✓		

Would you add or remove any project level agile characteristics? If so please explain why.

Do you of any further comments about the project level agile characteristics?

UE04

3.1 STAGE 3: ORGANIZATIONAL READINESS FOR AGILE LEVEL 1 (PAGES 18-25)

	strongly disagree	slightly disagree	neither disagree nor agree	slightly agree	strongly agree
UNDERSTANDABILITY					
For the topics listed below designate the degree to which you agree that they are understandable					
The objective or theme of Agile Level 1					✓
The agile practices and concepts identified in Agile Level 1					✓
The sample indicators and questions related to each agile practice or concept					✓
LEVEL OF DETAIL					
To what extent do you agree that the material about Agile Level 1 is presented at a sufficient level of detail?					✓
PRACTICALITY					
One of our objectives is to make sure that the Agile Levels are practical and can be used in industry. In light of this to what extent would you agree that if the agile practices and concepts presented in agile level 1 were used during the project that they would enhance the overall communication and collaboration?					✓
To what extent do you agree that the results of the questionnaires (indicators) legitimately reflect the readiness of the organization to adopt the agile practices and concepts identified in agile level 1?				✓	
EFFECTIVENESS					
To what extent do you agree that the agile practices and concepts identified in Agile Level 1 are sufficient enough to achieve the objective of this agile level which is enhancing communication and collaboration?				✓	
To what extent would you agree that the first level of agility should focus on enhancing communication and collaboration?			✓		
RELEVANCE					
To what extent do you agree that the agile practices in Agile Level 1 are relevant to their associated agile principals?					
Collaborative planning					✓
Collaborative teams					✓
Empowered and motivated teams					✓
Coding standards			✓		
Knowledge-Sharing tools			✓		
Task volunteering not task assignment				✓	
For each of the agile practices and concept in this level, to what extent do you agree that the sample indicators and questions are relevant and correctly linked to their respective agile practice or concept?					
Collaborative planning					✓
Collaborative teams					✓
Empowered and motivated teams					✓
Coding standards					✓
Knowledge-Sharing tools					✓
Task volunteering not task assignment					✓

Would you add or remove any agile practices or concepts to this agile level? If so please explain why.

Do you of any further comments about agile level 1

3.2 STAGE 3: ORGANIZATIONAL READINESS FOR AGILE LEVEL 2 (PAGES 26-34)

DEOL

	strongly disagree	slightly disagree	neither disagree nor agree	slightly agree	strongly agree
UNDERSTANDABILITY					
For the topics listed below designate the degree to which you agree that they are understandable.					
The objective or theme of Agile Level 2					✓
The agile practices and concepts identified in Agile Level 2					✓
The sample indicators and questions related to each agile practice or concept					✓
LEVEL OF DETAIL					
To what extent do you agree that the material about Agile Level 2 is presented at a sufficient level of detail?					✓
PRACTICALITY					
One of our objectives is to make sure that the Agile Levels are practical and can be used in industry. In light of this to what extent would you agree that if the agile practices and concepts presented in agile level 2 were used during the project that they would better ensure delivering software early and continuously?					✓
To what extent do you agree that the results of the questionnaires (indicators) legitimately reflect the readiness of the organization to adopt the agile practices and concepts identified in agile level 2?					✓
EFFECTIVENESS					
To what extent do you agree that the agile practices and concepts identified in Agile Level 1 are sufficient enough to achieve the objective of this agile level which is delivering software early and continuously?					✓
To what extent would you agree that the ~~first~~ 2nd level of agility should focus on delivering software early and continuously?				✓	
RELEVANCE					
To what extent do you agree that the agile practices in Agile Level 2 are relevant to their associated agile principals?					
Evolutionary requirements					✓
Continuous delivery (incremental- iterative development)					✓
Planning get different levels				✓	
Software configuration management			✓		
Tracking the iteration progress through working software					✓
No Big Design Up Front				✓	
For each of the agile practices and concept in this level, to what extent do you agree that the sample indicators and questions are relevant and correctly linked to their respective agile practice or concept?					
Evolutionary requirements					✓
Continuous delivery (incremental- iterative development)					✓
Planning get different levels					✓
Software configuration management					✓
Tracking the iteration progress through working software					✓
No Big Design Up Front					✓

Would you add or remove any agile practices or concepts to this agile level? If so please explain why.

Do you of any further comments about agile level 2

UEOV

3.3 STAGE 3: ORGANIZATIONAL READINESS FOR AGILE LEVEL 3 (PAGES 35-41)

	strongly disagree	slightly disagree	neither disagree nor agree	slightly agree	strongly agree
UNDERSTANDABILITY					
For the topics listed below designate the degree to which you agree that they are understandable					
The objective or theme of Agile Level 3					✓
The agile practices and concepts identified in Agile Level 3					✓
The sample indicators and questions related to each agile practice or concept					✓
LEVEL OF DETAIL					
To what extent do you agree that the material about Agile Level 3 is presented at a sufficient level of detail?				✓	
PRACTICALITY					
One of our objectives is to make sure that the Agile Levels are practical and can be used in industry. In light of this to what extent would you agree that if the agile practices and concepts presented in agile level 3 were used during the project that they would aid in the production of quality working software?					✓
To what extent do you agree that the results of the questionnaires (indicators) legitimately reflect the readiness of the organization to adopt the agile practices and concepts identified in agile level 3?				✓	
EFFECTIVENESS					
To what extent do you agree that the agile practices and concepts identified in Agile Level 3 are sufficient enough to achieve the objective of this agile level which is producing quality software?				✓	
To what extent would you agree that the ~~first~~ 3rd level of agility should focus on producing quality software?				✓	
RELEVANCE					
To what extent do you agree that the agile practices in Agile Level 3 are relevant to their associated agile principals?					
Risk driven iterations		✓			
Continuous improvement				✓	
Self-organizing teams				✓	
The use of true object oriented design and construction		✓			
Continuous integration				✓	
Maintaining the list of all remaining features				✓	
Unit tests				✓	
For each of the agile practices and concept in this level, to what extent do you agree that the sample indicators and questions are relevant and correctly linked to their respective agile practice or concept?					
Risk driven iterations				✓	
Continuous improvement				✓	
Self-organizing teams					
The use of true object oriented design and construction		✓		✓	
Continuous integration					✓
Maintaining the list of all remaining features				✓	
Unit tests				✓	

Would you add or remove any agile practices or concepts to this agile level? If so please explain why.

Do you of any further comments about agile level 3

DEUL

3.4 STAGE 3: ORGANIZATIONAL READINESS FOR AGILE LEVEL 4 (PAGES 42-48)

	strongly disagree	slightly disagree	neither disagree nor agree	slightly agree	strongly agree
UNDERSTANDABILITY					
For the topics listed below designate the degree to which you agree that they are understandable					
The objective or theme of Agile Level 4					✓
The agile practices and concepts identified in Agile Level 4					✓
The sample indicators and questions related to each agile practice or concept					✓
LEVEL OF DETAIL					
To what extent do you agree that the material about Agile Level 4 is presented at a sufficient level of detail?					✓
PRACTICALITY					
One of our objectives is to make sure that the Agile Levels are practical and can be used in industry. In light of this to what extent would you agree that if the agile practices and concepts presented in agile level 4 were used during the project that they would become more responsive to change?					✓
To what extent do you agree that the results of the questionnaires (indicators) legitimately reflect the readiness of the organization to adopt the agile practices and concepts identified in agile level 4?					✓
EFFECTIVENESS					
To what extent do you agree that the agile practices and concepts identified in Agile Level 1 are sufficient enough to achieve the objective of this agile level which is responding to change through multiple levels of feedback?				✓	
To what extent would you agree that the ~~first~~ 4th level of agility should focus on responding to change through multiple levels of feedback?					✓
RELEVANCE					
To what extent do you agree that the agile practices in Agile Level 4 are relevant to their associated agile principals?					
Client driven iterations					✓
Continuous customer satisfaction feedback					✓
Reflect and tune process				✓	
Smaller and more frequent releases					✓
Adaptive planning					✓
Daily progress tracking meetings			✓		
Agile documentation (from agile modeling)			✓		
User stories			✓		
For each of the agile practices and concept in this level, to what extent do you agree that the sample indicators and questions are relevant and correctly linked to their respective agile practice or concept?					
Client driven iterations					✓
Continuous customer satisfaction feedback					✓
Reflect and tune process					✓
Smaller and more frequent releases					✓
Adaptive planning					✓
Daily progress tracking meetings				✓	
Agile documentation (from agile modeling)					✓
User stories					✓

Would you add or remove any agile practices or concepts to this agile level? If so please explain why.

Do you of any further comments about agile level 4

DE02

3.5 STAGE 3: ORGANIZATIONAL READINESS FOR AGILE LEVEL 5 (PAGES 49-55)

	strongly disagree	slightly disagree	neither disagree nor agree	slightly agree	strongly agree
UNDERSTANDABILITY					
For the topics listed below designate the degree to which you agree that they are understandable					
The objective or theme of Agile Level 5					✓
The agile practices and concepts identified in Agile Level 5					✓
The sample indicators and questions related to each agile practice or concept					✓
LEVEL OF DETAIL					
To what extent do you agree that the material about Agile Level 5 is presented at a sufficient level of detail?					✓
PRACTICALITY					
One of our objectives is to make sure that the Agile Levels are practical and can be used in industry. In light of this to what extent would you agree that if the agile practices and concepts presented in agile level 5 were used during the project that they would establish a true Agile development environment?					✓
To what extent do you agree that the results of the questionnaires (indicators) legitimately reflect the readiness of the organization to adopt the agile practices and concepts identified in agile level 5?					✓
EFFECTIVENESS					
To what extent do you agree that the agile practices and concepts identified in Agile Level ~~1~~ 5 are sufficient enough to achieve the objective of this agile level which is establishing a true agile development environment?					✓
To what extent would you agree that the ~~first~~ 5th level of agility should focus on establishing a true agile development environment?					✓
RELEVANCE					
To what extent do you agree that the agile practices in Agile Level 5 are relevant to their associated agile principals?					
Low process ceremony					✓
Agile project estimation				✓	
Paired programming				✓	
Test-driven development				✓	
For each of the agile practices and concept in this level, to what extent do you agree that the sample indicators and questions are relevant and correctly linked to their respective agile practice or concept?					
Low process ceremony					✓
Agile project estimation					✓
Paired programming				✓	
Test-driven development				✓	

Would you add or remove any agile practices or concepts to this agile level? If so please explain why.

Do you of any further comments about agile level 5

ULOV

APPENDIX B: INDICATOR AGGREGATION (PAGES 60-64)

	strongly disagree	slightly disagree	neither disagree nor agree	slightly agree	strongly agree
UNDERSTANDABILITY					
For the topics listed below designate the degree to which you agree that they are understandable					
The directions on how to use the automated method					✓
How to compute a weight for each indicator					✓
How to compute the weighed interval					✓
How to calculate optimistic and pessimistic results					✓
How to translate the results into a nominal score					✓
LEVEL OF DETAIL					
To what extent do you agree that the material about the indicator aggregation was presented with a sufficient level of detail?				✓	
PRACTICALITY					
One of our objectives is to make sure that our proposed method for indicator aggregation is practical and can be used in industry. In light of this to what extent would you agree that this approach to indicator aggregation can be used practically?					✓
RELEVANCE					
If this approach is used to aggregate a set of indicators, to what extent would you agree that the results would legitimately reflect a valid and relevant outcome?					✓
EFFECTIVENESS					
To what extent do you agree that this approach to indicator aggregation is a sufficient and valid one in aggregating the various sets of indicators throughout the process framework?					✓

Do you of any further comments about the method of indicator aggregation?

____

____

____

____

DF02

Per our prior email, I'm going to type my comments in this email that I marked on the various pages. I'll include page numbers where you had them.

Matrix of the 5 Agile Levels (first section, the big grid): I don't like that you want these to be "risk-driven iterations." Boehm proposed the spiral model as a risk-driven approach to software development. Tom Gilb said we should do the "juicy bits first." I've also argued that agile is about the juicy bits first. Risk is considered but is not primary. I use "value-driven iterations." Other items I'd place different but I suspect those are much more a matter of opinion that "risk-driven".

Page 6: You list whether the org has problems going over time or budget in the first box. I think there are many opportunities beyond just those. I was with two clients in the last two weeks who each listed "innovation" as something they are hoping to gain from agile. So, a lack of innovation, a lack of product market responsiveness, etc. are among the other problems that agile solves.

Page 6: In the para starting "in order..." I circled the word "trend" in the next-to-last line. "history" is better than "trend" because there may be no trend; it just may be what exists.

Page 6: You say (at the bottom) that D should be answered by a developer. You should defined "developer." I use that term for anyone on the development side of a project. I got the sense that you mean programmer.

Page 7: Factor 3: This doesn't consider that the effort may be self-funding. This is sometimes the case. I've seen companies where process improvement departments get next year's budget based on evidence of improvements they made this year.

Page 9: I don't like how DC_M16 and M17 are worded. The "failure would not result..." is awkward. I realized later that you did this so that the V-Z values are consistent but it leads to some awkward sentences. Yet another reason that the analysis of responses should be by software (either online questionnaire or scanned and OCRd results).

Page 9: I didn't like DC_D1: "There are many areas...that always...." How about "some that frequently"? "many that always" seems too unlikely to occur.

Page 11: You introduce the PPAL but I think many companies are going to be interested in a Company PAL. I'm also not wild about the world "potential." Would "Target" be a better word?

Page 11: Second para starts with "the rational" but should be "the rationale" (add the e)

12: For assessment level a, it seems like there should be a principle about whether the customer is also able to work with the development team (not just committed to doing so).

14: AC_C3: says "that the contractor should..." It seems like this s/b the "development team" not "contractor"

15: In AC_C13 you ask the Customer if they are immediately accessible to the team. I'd also like to know the team's view on this. I suspect teams and customers will differ on "immediately accessible!"

16: AC_A2: The "within driving distance" and "within the same city/area" may be swapped in some geographic regions (e.g., parts of CA). I wonder if a time-based set of answers would be better? I'm not sure, though.

16: For AC_A5 and AC_A6 it would be better to have questions that get at whether the team has the right type of people rather than to go there directly with Cockburn Level 2/3.

OE02

17: "that need to be assessing" s/b "assessed" I won't point out typos in general but the one above and this one will be hard to find otherwise since they're correct spellings.

22: OR1_D1 is too direct. It says what is needed rather than how to assess it. Give the developer some questions that you can use the answers from to tell if the manager is collaborative rather than asking directly.

23: OR1_D16. I circled the word "usually" in this one and noted that sometime you included usually in the question but not always. Perhaps you've thought this through thoroughly on each but some of the wording distinctions seem random. And when "usually" isn't there is "always" implied? (e.g., on OR1_D17)

23: OR1_D20. There's this amazing love of wikis (especially) and blogs (somewhat) among agilists and yet they hate all other tools. I like wikis and use them daily but I don't know that use of a wiki is on a par with having a coding standard (D21). D19 seems fine with D20 in this case.

24: OR1_M2 asks if you prefer team work over individual work. Personally, I prefer individual work 100-1 over team work yet I think I'm a great manager for teams because I don't let my personal preference for style influence how I have teams work. I don't like this question but some of that is undoubtedly me.

24: I noticed on this page that you're inconsistent throughout between center and left-justifying these statements.

24: For OR1_M9, M10, and M17 you start with "you believe that..." This seems like it should be in front of many more statements or none. I don't see what you lose by dropping it.

24: Similarly, for OR1_m12 I'd drop "would" and I'd change OR1_M13 to "Your subordinates have unregulated..."

27: in the last row on the page I'd change it to "the organization has AND USES tools for...

28: In People | Management | Stress you say "overlooking" but "overseeing" is probably a better word. Overlooking usually means closer to "ignoring."

30: I'd change OR2_D2 to start with "You are willing to start..." instead of "You can start..." since I "can" but may not be willing to.

32: By this page I noted, "It seems like parts of this assessment need to be given 1-2 levels higher in the organization than the group you are assessing." I think that many managers will say things that may or may not be supported up the organization.

36: People | Developers | Buy in says "whether or not the employees feel comfortable working as self-organizing teams." It seems like this s/b "whether or not employees ARE working as self-organizing teams."

37: Continuous Integration second row says "has the tools"? Why are these tools so significant? You didn't ask earlier whether the organization has a wiki or could set one up--you just focused on whether it was used. I'd hardly say "a company can't make it to the third level because they don't yet have the tools." Buying a tool will be the least of most companies problems.

45: User Stories | Process |Requirements | Regulations: I've never seen a regulation saying requirement has to be in a certain form. I've used stories, for example, for ISO 9001 projects.

DE02

46: OR4_D10: I don't think the daily standup is a "progress update." I'm not willing to meet for a progress update. I am willing to meet for a quick synchronization meeting and to check in with my colleagues.

52: OR5_D4: You say "project estimates" but perhaps "Feature estimates" may be better. I don't ever want anyone to estimate a whole project. I want to estimate the features and then figure out how fast we progress. From that we can derive a project estimate.

61: You have a link to https://www.oracomputer.com/ee but there's nothing at that url.

62: It looks like you edited parts of your example. At the bottom in text you add 0+17+28=41 but use 13 not 17 in the table. The same is true for the Optimistic Result right below (28 and 20 are used)

63: You say "if they do not then obtain an average and then place that average in its necessary nominal range." You may want to show a second example and show this.

Thanks again for sharing your research with me. I hope I get to see it in use at Yahoo. Please let me know if there's further feedback I can provide.

Vita

Ahmed Sidky is a senior agile consultant with Tangible Software. He graduated as Valedictorian with a Bachelor's degree in Computer Science from the Modern Science and Arts (MSA) University in Cairo, Egypt. While working as an Internet Solution Developer for one of the leading corporations in Egypt, he received the award for the Best Creative Solution for that year. With his research focused on Requirements Engineering, he earned a Masters degree in Software Engineering from Virginia Tech. Ahmed's research interests then moved towards Agile Software Development Methodologies and he is completing his doctorate in the Spring of 2007 in that field. His latest research is a process framework for the adoption of agile practices known as the Agile Adoption Framework.