Agile on the rise

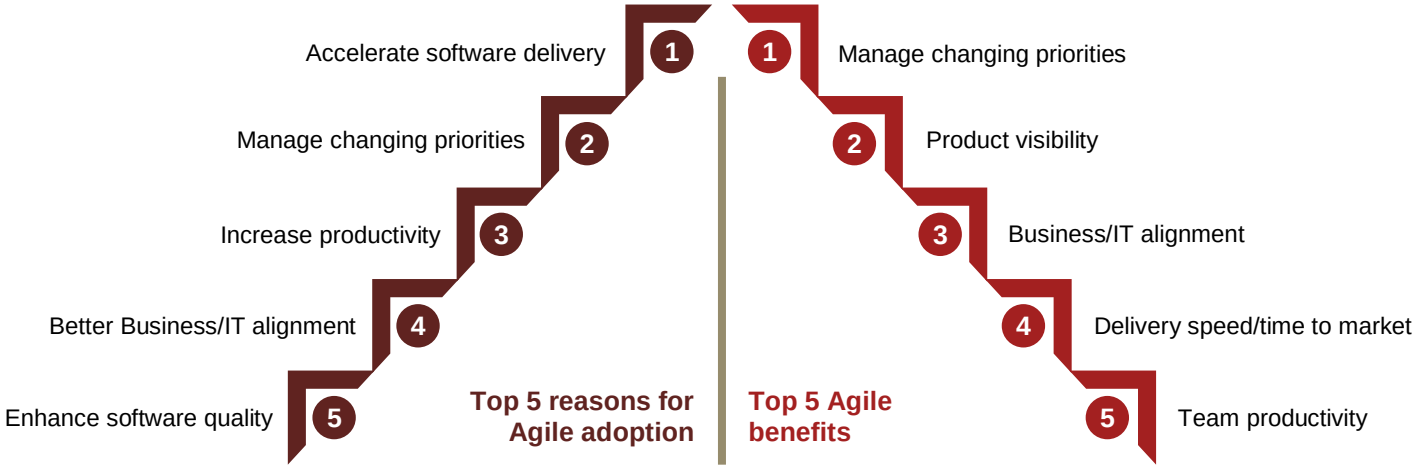
Integrating effective controls into Agile environments

pwc

Agile adoption in today's market

Organizations are increasingly adopting **Agile methodologies for IT development and ongoing maintenance.** In fact, the Project Management Institute's 2018 Pulse of the Profession survey reports that 87% of surveyed organizations used some form of Agile practices in the past year[1]. Organizations face an ever-changing business environment including a drive to develop new and innovative digital solutions to keep ahead of their competitors. Agile practices can accelerate software delivery, raise IT productivity, improve software quality, increase customer and employee satisfaction, and help organizations quickly adapt technology as business priorities change (Figure 1)[2].

Figure 1:



Yet, despite business and IT embracing Agile, many organizations still have risk, compliance, and assurance functions that do not have a seat at the table resulting in non-controlled, non-compliant and less effective Agile adoption. But it doesn't have to be this way. Through understanding some of the nuances and misconceptions, risk management teams can offer valuable feedback and support as an organization takes the plunge into Agile.

A common misconception: Agile inherently weakens controls

Stories abound of renegade teams that use their Agile approach as justification for poor formality or discipline. For example, teams may forgo formal production of mandatory project documentation or bypass required approval or gating steps in the name of the speed required for Continuous Integration/Continuous Delivery (CI/CD). These stories can accumulate into a broader misconception that Agile is unsuitable in a mature internal controls environment for risk management functions or, for engineers, can translate to "you're slowing me down from doing my job".

Alternatively, when an organization has successfully gone Agile, we've seen compliance functions attempt to rapidly retrofit old-school controls into the development cycle leading to an erosion of Agile productivity gains.

However, if designed and implemented well Agile methodologies along with DevOps tools and processes can address internal control objectives in a lean and efficient manner and enhance the overall control environment. This requires the organization to have an eye to scalability, sustainability, and to use the full power of tools and functions at an organization's' disposal.

[1] "Success in Disruptive Times," PMI's Pulse of the Profession, 2018 [2] Source: VersionOne 12th Annual State of Agile report

Agile transformation nuances

There are significant differences in how organizations adopt Agile. These factors influence the nature and design of controls within an enterprise's Agile environment. Differences relate to three key factors:

Methodology

Agile is an umbrella term for multiple methodologies that feature short phases of work, collaboration, and iterative improvement to plans and solutions. Organizations may adopt their own methods, select from a range of published methodologies (such as Scrum or Extreme Programming (XP)), or blend elements of traditional waterfall approaches with iterative techniques ("hybrid").

01

Scaling

Agile is often associated with small teams working on a single product. However, organizations are also applying Agile principles to large, multi-team initiatives, including their entire enterprise delivery model (referred to as Agile-at-scale). The maturity of different organizations (across process, tools, etc.) will also vary despite having similar scaling models.

02

DevSecOps

The adoption of DevOps and ultimately DevSecOps in some organizations is often closely associated with the adoption of Agile. DevSecOps uses two distinctive concepts: (1) the confluence of software development, information security and IT operations groups and (2) the use of automation in those activities.

03

Understanding how the organization is embracing Agile relative to these factors will help risk, compliance and assurance functions scope and design appropriate controls.

Case study 1: When adopting Agile fails at a Fortune 500 bank

Teams across a Fortune 500 bank began independently piloting Agile. This was met with initial excitement, but without a robust adoption roadmap, the organization hit several pain points:

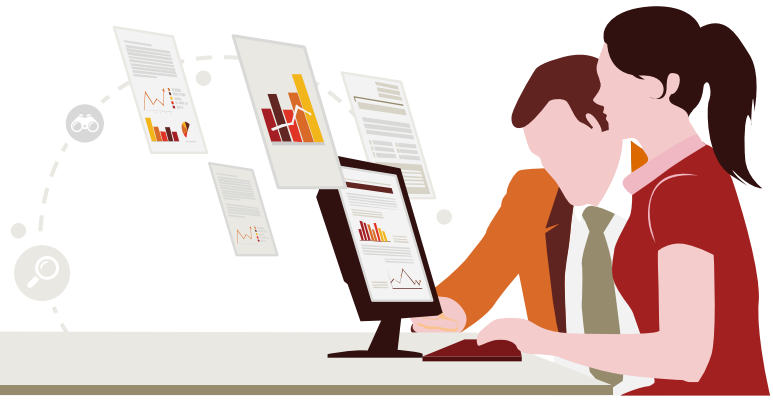
- Enthusiastic Agile adopters overlooked required regulatory compliance evidence, leading to penalties
- Other Agile teams produced additional material to comply with legacy requirements leading to re-work and reduced productivity
- Integration was delayed and flawed because there was no mechanism for teams to coordinate with non-Agile teams (eg. security and architecture mainframe teams)
- Without standardization or guidance, teams developed Agile anti-patterns reducing effectiveness and leading to business aversion to Agile

Integrating controls into an Agile environment

When considering a controls landscape, the following controls and standards requirements are important to integrate within an Agile environment:

- **SOX Reporting (SEC)** - Internal controls reporting in the company's SEC regulatory filings
- **SOC Reporting (Customers)** - External controls reporting to the company's customers
- **Company Quality and Control Standards** - Internal expectations around quality and consistency of code development in meeting the business needs, e.g. efficiency and speed to market
- **Regulatory and other Compliance Risk** - External expectations related to the production and submission of data to regulators or the meeting of other regulatory needs

Once the breadth of required control objectives are understood, organizations should consider designing specific controls within the process. As a general rule, the *control objectives* that have always applied to application development and operations remain the same; however, individual *control activities* may differ in an Agile environment.



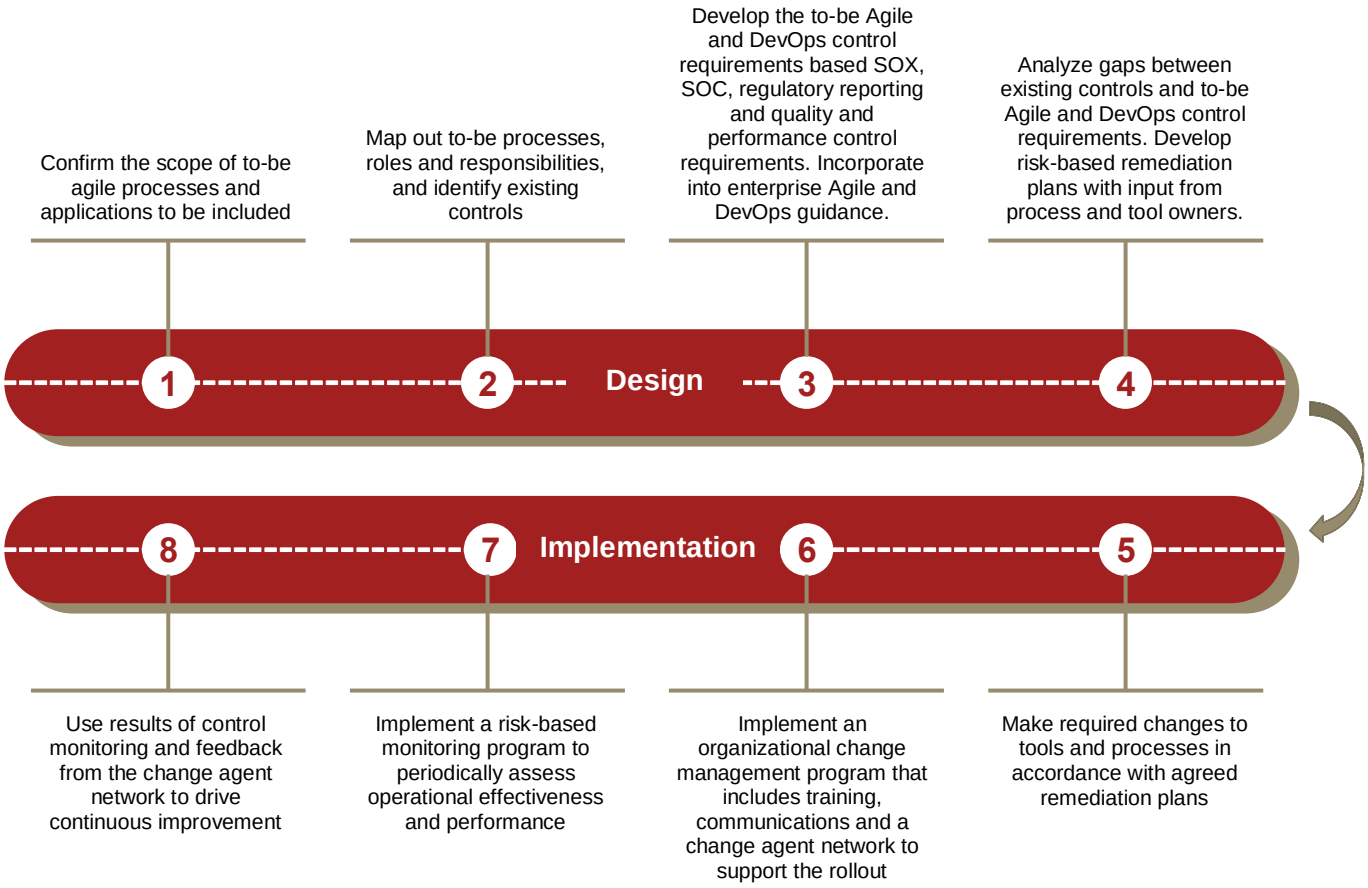
Effective control activities may be embedded in an organization's Agile methodology and/or DevOps tools. Below are three illustrative examples of how control activities can be integrated in support of important control objectives:

Control objective	Concern	How comfort can be gained through new practices and evidence
1 **Segregation of Duties**	Historically, distinct teams would complete work in separated development, test, and production environments. This segregation of duties helped confirm that only approved changes were applied to production. In some DevOps scenarios, there may now only be one team.	• Ability to deploy changes to production code is restricted to automated tools. • Automated release management tools are configured such that: - Automated testing must be completed satisfactorily prior to deployment - Configuration access is restricted to approved administrators - Logging of appropriate events is enabled and monitored • An additional consideration exists around the automated testing tools. Management needs processes in place to make sure test scripts are kept up to date with changes and that an agreed level of coverage is achieved.
2 **Traceability**	Historically, auditors could confirm due diligence was taken during system development activities by reconciling formally documented and approved project charters, business requirements, design documents, and user acceptance test reports. Agile teams may now balk at producing similar artifacts, citing the Agile Manifesto value of "working software over comprehensive documentation".	• "Requirements", typically in the form of Epics, Features and Stories, are stored in a tool such as JiRA, Rally, or Team Foundation Server (TFS) • Traceability is then established to track each story through the tool chain to show approvals, testing results and ultimately its release into production • Evidence and approvals are retained in the tools to allow for auditability. For example authorized product owners may have to sign-off user stories using a checkbox approval at the end of a sprint (requires role based access to be defined and for appropriate business representation to be integrated into the production lifecycle).
3 **Code quality and reliability**	Historically, a distinct testing (or QA) team would be responsible for verifying the quality of software through manual testing routines. Agile teams may now combine development and testing duties, and advocate a 'fail fast' approach.	QA can be performed through a number of means: • Establishing quality expectations upfront (e.g. definition of release, definition of ready, definition of done) • Code Scanning - integrated vulnerability and performance scans executed upon code check-in • Continuous Integration - automated integration testing run at code check in • Automated testing - use of integrated code test tools to run repeatable tests covering code identifying issues and vulnerabilities • Robust event logging - DevOps tools often provide robust activity logs detailing user and tool activity.

The contents of the table above is far from a complete list, and examples are purely illustrative. Each organization should consider the specific controls and activities it requires. Fortunately, many mature controls frameworks (e.g. COBIT5) already embrace Agile principles.

Steps to establishing a right-sized control framework in an Agile environment

When organizations begin to adopt Agile, risk, compliance and assurance teams may be unsure where to begin. The following steps can serve as a guideline:



Case study 2: Building controls into the process at a top five healthcare payer

Agile was being rolled out to a subset of applications at a top five healthcare payer. There was a concern that audits may fail once applications or enhancements were promoted to production. Executive leadership took the initiative to ensure controls were being built into Agile processes.

- A gap analysis was performed against the expectation of SOX, Internal Audit, SOC, Security, and other regulatory requirements
- Control gaps that were identified were closed and controls training performed
- As Agile scaled compliance with the controls framework was limited due to lack of awareness, awareness training and communication was undertaken to promote the importance of adherence to the controls framework
- Ongoing monitoring was then established to help drive compliance with the controls
- Better integration with the DevOps teams was promoted, as DevOps was seen as key to maximizing the benefits from Agile

Leading practices in controls identification and design for Agile and DevOps

As design and implementation steps are executed, organizations should integrate leading practices wherever possible. These help to design effective and cost-efficient controls and to sustain them over time. Based on substantial experience designing controls for Agile and DevOps environments, PwC has identified seven leading practices:

1. Leading organizations launch Agile transformation initiatives to develop their Agile capabilities, which often includes creating a *Center of Excellence* to sustain adoption and oversee continuous improvement of Agile methods, tools and skills. A dedicated, and integrated controls workstream will help ensure controls are integrated, efficient, and effective.

2. A documented enterprise approach for Agile projects (for example a 'playbook' containing methodology, protocols, approved tools, etc.) helps teams produce repeatable results and consistent evidence, and makes adoption and collaboration more effective.

3. Internal control activities should be embedded within an enterprise's Agile approach. In general, these should not be obvious additions (overhead) to the Agile team's guidance. Teams can capture more formal controls descriptions in other repositories (e.g. GRC tools).

4. Adopting the following leading practice internal controls design principles, where possible, will aid Agile adoption:
   a. Automated rather than manual controls
   b. Preventative rather than detective controls
   c. Integrated controls that are natural and beneficial to the business process flow, rather than adding overhead

5. Mature Agile disciplines typically produce auditable evidence of management control. As it may not be immediately intuitive or instantly recognizable, leverage and consult with experienced practitioners for guidance.

6. This whitepaper has largely focused on controls in the Agile development and maintenance processes. The products created or modified by Agile teams must also incorporate effective security and application controls. Embed triggers and guidance for these considerations within the Agile approach.

7. Project quality audits should place more emphasis on observing Agile roles, behaviors, and practices in action rather than assessing documentation after the fact. (For further reading, refer to 'Internal audit: Thinking differently in an agile organization' and '*Agile Project Delivery Confidence*' whitepapers.)

Changing controls activities to enable the rise of Agile

As the use of Agile becomes pervasive, all risk, compliance, and assurance executives need to embrace how these highly effective methods can co-exist with effective controls. With a sufficient understanding of the Agile environment and leading controls development practices, risk professionals can take the right steps to integrate controls that protect against risk and non-compliance without compromising much needed agility.

Contact us

For a deeper conversation about integrating effective controls into Agile environments, please contact us:

Primary contributors

Mike Shipham
Director
M: (312) 206-6158
michael.a.shipham@pwc.com

Matt Bonser
Director
M: (415) 518-9895
matthew.p.bonser@pwc.com

Supporting contributors

Donna DiGiacomo
Principal
M: (646) 471-7102
donna.digiacomo@pwc.com

Gary Harvett
Managing Director
M: (973) 236-4399
gary.harvett@pwc.com

Zach Sachen
Principal
M: (801) 534-3878
zach.sachen@pwc.com

Kara Finley
Principal
M: (678) 362-8798
kara.finley@pwc.com

Jerry Stone
Partner
M: (410) 659-3630
jerry.stone@pwc.com

Niket Desai
Principal
M: (312) 298-3838
niket.desai@pwc.com