Project 2013

IN DEPTH

Scott Daley

Microsoft

Project 2013

IN DEPTH

Scott Daley

que®

800 East 96th Street
Indianapolis, Indiana 46240

PROJECT 2013 IN DEPTH

ISBN-13: 978-0-7897-5095-2
ISBN-10: 0-7897-5095-3

Library of Congress Cataloging-in-Publication Data:

Printed in the United States of America

First Printing: July 2013

Trademarks

All terms mentioned in this book that are known to be trademarks or service marks have been appropriately capitalized. Que Publishing cannot attest to the accuracy of this information. Use of a term in this book should not be regarded as affecting the validity of any trademark or service mark.

Warning and Disclaimer

Every effort has been made to make this book as complete and as accurate as possible, but no warranty or fitness is implied. The information provided is on an "as is" basis. The authors and the publisher shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this book.

Bulk Sales

Que Publishing offers excellent discounts on this book when ordered in quantity for bulk purchases or special sales. For more information, please contact

U.S. Corporate and Government Sales

1-800-382-3419

corpsales@pearsontechgroup.com

For sales outside of the U.S., please contact

International Sales

international@pearsoned.com

Associate Publisher
Greg Wiegand

Executive Editor
Loretta Yates

Development Editor
Sondra Scott

Managing Editor
Sandra Schroeder

Project Editor
Seth Kerney

Copy Editor
Geneil Breeze

Indexer
Erika Millen

Proofreader
Sarah Kearns

Technical Editor
Alan Wright

Publishing Coordinator
Cindy Teeters

Designer
Anne Jones

Compositor
Jake McFarland

CONTENTS AT A GLANCE

CONTENTS

IV Tailoring Microsoft Office Project 2013 to Your Needs

18 Managing Project Files Locally and in the Cloud 549

19 Formatting Views 569

ABOUT THE AUTHOR

Scott Daley has been updating *Project In Depth* since the Project 2010 edition. He has been working with Project and Project Server for over a decade as a consultant, a Microsoft employee, and an independent consultant. Scott believes that project management done well is a highly underrated discipline.

This book was written with the intent to explain more than just what Project does or why it does it, and to go beyond the standard narratives surrounding Project. Scott has seen Project put to use managing many different kinds of projects, and hopes that this book will clarify some of the reasons why these efforts can succeed or fail.

ABOUT THE TECHNICAL EDITOR

Alan Wright has worked professionally in and around IT for nearly 10 years. He has provided enterprise-level support in the Detroit, Michigan, area and continues to provide software and hardware support for small business and residential users. He holds several certifications from CompTIA and Microsoft and enjoys working with technology and teaching others how they can make technology work for them.

DEDICATION

This book is dedicated to all the deliberate and accidental project managers who, in the end, just want to deliver.

ACKNOWLEDGMENTS

Scott would like to thank QuantumPM for the contribution they made to this edition of *Project In Depth* by writing the 2007 edition.

Scott would also like to acknowledge the hard-working staff at Que Publishing who have helped make this book what it is today. He would especially like to thank Executive Editor Loretta Yates. It has been his pleasure working with her over the years.

WE WANT TO HEAR FROM YOU!

As the reader of this book, *you* are our most important critic and commentator. We value your opinion and want to know what we're doing right, what we could do better, what areas you'd like to see us publish in, and any other words of wisdom you're willing to pass our way.

We welcome your comments. You can email or write to let us know what you did or didn't like about this book—as well as what we can do to make our books better.

Please note that we cannot help you with technical problems related to the topic of this book.

When you write, please be sure to include this book's title and author as well as your name, email address, and phone number. We will carefully review your comments and share them with the author and editors who worked on the book.

Email: feedback@quepublishing.com

Mail: Que Publishing
ATTN: Reader Feedback
800 East 96th Street
Indianapolis, IN 46240 USA

READER SERVICES

Visit our website and register this book at www.quepublishing.com/register for convenient access to any updates, downloads, or errata that might be available for this book.

This page intentionally left blank

3

MICROSOFT PROJECT AND THE PROJECT MANAGEMENT DOMAIN

This chapter is about using Microsoft Project within the domain of project management. It identifies accepted standards and methodologies that are in use within the field and provides some examples of how they are used with Project.

History of Project Management

When Patrick Henry said, "I know of no way of judging the future but by the past," he could have been talking about project management. When faced with projects that have never been done before, all project managers can do is look at what has come before them.

Although project management has been practiced for thousands of years, evidenced by the Egyptian and Roman dynasties, modern project management can be traced back to the late nineteenth century and the rise in large-scale government projects and growing technological advancements. Fredrick Taylor, the Father of Scientific Management, applied scientific reasoning to analyzing and improving labor, and Henry Gantt studied management of Navy ship construction during World War I. Gantt's use of charts, task bars, and milestone markers is still practiced today, and they bear his name. One of the major projects that brought detailed project planning, controlling, and coordination to the forefront was the Hoover Dam project, which involved $175 million dollars, six different companies, a major worksite with no existing infrastructure, and approximately 5,200 workers. The project was brought in under budget and ahead of schedule.

After developments in project management during two World Wars and the growing Cold War, major changes to project management were brought about with the launch of Sputnik. Fearful that the United States was falling behind in the race to space, the United States introduced several major programs to focus on science and exploration. Several agencies, including the Advanced Research Project Agency, a high-level research and development program that later became DARPA, and NASA were founded. These agencies led the way in the development of project management.

Two other major developments for project management to grow out of this period were the Critical Path Method (CPM) and the Program Evaluation and Review Technique (PERT). CPM was devised by Du Pont and Remington Rand for use with the UNIVAC-1 computer mainframe. PERT was invented by the Program Evaluation Branch of the Special Projects office of the U.S. Navy, for use with the POLARIS missile program, and was also used on the Apollo program for NASA. CPM/PERT gave managers more control over extremely large and complex projects, but could only be calculated within large mainframe computer systems and were used mainly for government sector projects.

With the computer revolution of the 1980s and the move from mainframe computers to personal computers with the ability to multitask, project management software became more accessible to other companies. The Internet and networked systems only made project managers more efficient at controlling and managing the different aspects of their projects. More information on previously completed projects is available today than ever before, making the project manager's job of estimating the future by looking at the past easier than ever.

Exploring Project Management Industry Standards

Almost anyone can create a schedule with Project. Organizing that schedule into a logical flow of work, however, requires a solid understanding of how projects should be managed and decomposed into logical units. To understand project management, you must understand the standards and methodology behind it. Although Gantt Charts and other similar resources are used in almost all project management schedules, there are several different ways of using those resources.

This chapter discusses prominent industry standards often used to set a framework for building schedules. A variety of methodologies, team styles, and life cycles also are explored. The approach and techniques vary, but the software can still be used to support virtually any approach to scheduling that an individual or organization chooses to use.

Project Management Body of Knowledge (PMBOK)

The Project Management Institute, or PMI, is an internationally recognized organization that has developed standards for the domain of project management including standards for portfolio management, program management, project management, and Work Breakdown Structures. PMI has several hundred thousand members in more than 65 countries. It is widely recognized for its certification programs and continues to grow through a combination of volunteer efforts, certification programs, local chapter events, international seminars, and special interest groups.

The standards created by PMI are authored by a vast network of project management professionals who volunteer their time to create and update these standards on a regular basis. The standards

groups are from many different countries across the globe; they research topics and collaborate to bring together the latest thinking and techniques from their collective experience.

The PMI standard that is of primary importance for this chapter of the book is in its fifth edition and is known as "A Guide to the Project Management Body of Knowledge," also known as the PMBOK Guide. It is discussed in some detail in this chapter to help in understanding all the components that should be considered when creating a schedule.

Because PMI is a standards and certification organization, it does not prescribe methodologies or "how to" approaches; rather, it defines specific standards and offers certifications in the field of project management. The PMBOK provides a context for a way to do things, rather than the process that should be followed.

Inexperienced project managers often try to make their schedules follow PMBOK as if it were a recipe for success. This can lead them into traps and complexity that is not useful in completion of their projects. Instead, they should look to the PMBOK for support of the methodology and life cycle that they choose to follow.

The PMBOK Guide has established five process groups to define the project management process. These processes are as follows:

- **Initiating Process Group—**Defines and authorizes the project or a project phase
- **Planning Process Group—**Defines and refines objectives, and plans the course of action required to attain the objectives and scope that the project is to address
- **Executing Process Group—**Integrates people and other resources to carry out the project management plan
- **Monitoring and Controlling Process Group—**Regularly measures and monitors progress to identify variances from the project management plan so that corrective action can be taken when necessary to meet project objectives
- **Closing Process Group—**Formalizes acceptance of the product, service, or result and brings the project or a project phase to an orderly end

PMI makes a clear distinction between a *project plan* and a *project schedule*. The *plan* is a formal document that includes narrative on communication approaches, assumptions, deliverables, and execution of the project. The *schedule* is one component of the plan that focuses on the timeline for the activities to be performed. As scheduling tools become more sophisticated, they are gradually including more elements that used to reside only in the plan. Project Desktop still focuses on scheduling functions, but the server components have added capabilities to support more of the plan functions.

Projects are created and implemented in environments that are larger in scope than the projects themselves. All projects must have a beginning and an end, as shown by the Initiating and Closing process groups. In between, a project will be engaged continually with the other three process groups, as shown in Figure 3.1.

The PMBOK identifies nine knowledge areas that a project manager should consider throughout the entire life cycle of a project. Knowledge areas focus on a specific aspect of the overall domain and identify the elements that need to be considered to properly manage a project:

- **Project Integration Management—**This knowledge area looks at the processes and activities needed to identify, define, combine, unify, and coordinate the different actions within a project management process group.
- **Project Scope Management—**This knowledge area handles scope planning, scope definition, creating a WBS (decomposition of the scope into smaller components), scope verification, and scope control.
- **Project Time Management—**This knowledge area concerns five different steps: activity definition, activity sequencing, activity resource estimating, activity duration estimating, and schedule development.
- **Project Cost Management—**This knowledge area involves planning, estimating, budgeting, and controlling costs so a project can be finished within budget.
- **Project Quality Management—**This knowledge area determines policies, objectives, and responsibilities to meet a project's quality standards.
- **Project Human Resource Management—**This knowledge area helps organize and manage a project's team, the people necessary for the completion of the project.
- **Project Communications Management—**This knowledge area involves the processes that ensure timely generation, collection, distribution, storage, retrieval, and disposition of information.
- **Project Risk Management—**This knowledge area envelopes risk management planning, identification, analysis, responses, monitoring, and controlling of a project.
- **Project Procurement Management—**This knowledge area involves the processes necessary to purchase products, services, or results from outside the project team.
- **Project Stakeholder Management—**This knowledge area involves the processes necessary to manage the expectations of and results for the many people and things that can have a stake in the execution and outcome of your project.

The ten knowledge areas are specifically designed to work with the five process groups to identify possible areas for management within the scope of the project. When the two components are combined, they provide guidance for what elements should be considered at what time in a project.

In the context of the Project desktop, the key knowledge areas are scope, time, and cost. These components help you build the initial project schedule framework.

The emphasis for each knowledge area varies by phase of project; some are more important in one phase than another, but all of the nine are used throughout the project.

note

Do not confuse the PMBOK process groups with life cycle phases of projects. This is a common tendency when a project manager tries to decompose a project into logical components. Process groups pertain to all projects; life cycles vary by the type of project, the domain of the work, the complexity and timeframe, and many other factors. Details about phases are covered later in this chapter.

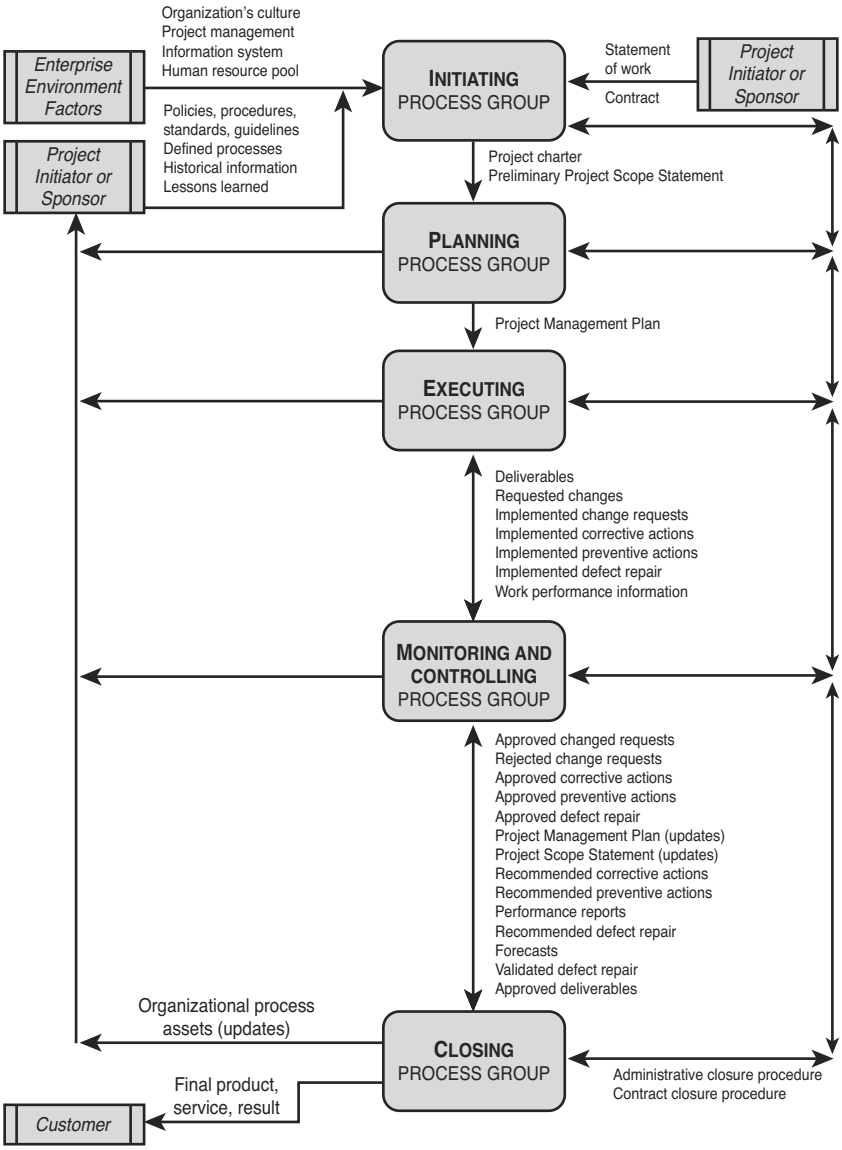

Figure 3.1
The relationship among the PMBOK process groups (taken from Figure 3.3, *PMBOK Guide, Fifth Edition*).

PRINCE2

PRINCE2, which stands for *Projects in Controlled Environments*, is a project management methodology developed by the United Kingdom government. It is in its second release and was originally known as the PRINCE technique. The first release was established in 1989 by the Central Computer and Telecommunications Agency (CCTA) as a standard for information technology project

management. Because of its success in IT, the methodology was republished about seven years later in a version that could be applied across many other disciplines. PRINCE2 was again updated in 2005 by the Office of Government Commerce (OGC), has become the standard for project management in the United Kingdom, and is now used in 50 other countries. You can become certified in the use of PRINCE2 at either one of two levels: Foundation and Practitioner.

PRINCE2 uses a simple four-step process to explain what each project needs, as shown in Figure 3.2. This process is explained in more detail using the following eight different processes, sometimes known as the Validation, Quality, Verification, and Approval steps:

- **Start-up—**This is when a project manager is chosen. The need for the project is defined and outlined as to how it will be executed.
- **Direction—**The project manager, who reports to the Project Board, is responsible for managing the details. The Project Board is responsible for the overall success of the current project and defines the direction in which the project will be heading.
- **Initiation—**The Project Initiation Document is prepared and submitted to the Project Board for approval and possible revision.
- **Stage Control—**During this stage, the project is broken down into several different manageable stages. The number of stages depends on the size and risk level of the project, and each stage must also plan for the succeeding stage. Before any new stage can begin, the current stage must be fully finished.
- **Stage Boundary Management—**At this stage, the Project Board must review the current stage and then develop the process for the next stage. It is only after the approval for the execution of the current stage and the planning of the next stage that the project can continue.
- **Planning—**This stage is used for deciding what products will be produced and what is required for their production. Then, estimates are made for cost, time, and any other resources, plus any risk analysis, activity scheduling, and process streamlining that is necessary for the project.
- **Product Delivery Management—**This is the production stage, where the project manager confirms that the right goods are being produced correctly and on schedule.
- **Closing—**After everything is finished, the project manager must perform a post-project review, which evaluates the outcome of the project. When this review is approved by the Project Board, the project is complete.

In addition to consideration of these standards and methods, project managers need to understand the environment in which they will be working before they create a schedule. They need to be aware of the various methodologies and approaches that can be used to help them (or confuse them, if they do not understand how and when the methodologies and approaches should be applied). The following section provides an introduction to this information.

note

There is no conflict between PMBOK and PRINCE2. They can be used together if a project manager chooses to use both. PRINCE2 is a methodology and focuses more on deliverables, whereas PMBOK is a standard and focuses on the process and knowledge areas. PRINCE2 establishes a Validation of Process (through a specific focus on deliverables and the activities around them), whereas PMBOK focuses on the processes used to manage the deliverables.

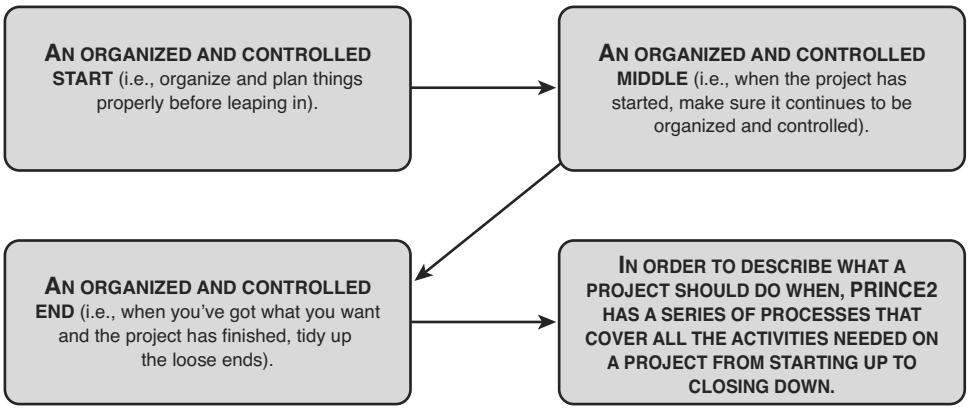

Figure 3.2 The PRINCE2 four-step process.

WBS, Phases and Control Points, Methodologies, and Life Cycles

note

Many of the preceding terms are used by project managers to describe the approach that is used to define and execute a project. Each of these has been explained in many other texts and references. Because the focus of this book is using Project, the topics are brought up here to provide context only; there is no attempt to provide the definitive use of any of the terms. Rather, the hope is that the reader can apply the concepts and techniques as appropriate when building a schedule. The work that needs to happen during a project's life does not automatically conform to a particular methodology.

Before building any schedule, the project manager must consider two key components: work decomposition (what work needs to be done, the Work Breakdown Structure or WBS) and managerial control (stages, phases, and life cycle requirements). The discipline used for either depends on the environment in which the project is executed, so the formality will vary, but both components must be considered. The tasks or activities and milestones (how the work will be accomplished) should not be defined in a project schedule until the WBS and control framework are determined. WBS helps the project manager set parameters around the scope of work to be done; the life cycle sets the controls in place for decisions during project execution. If these two components are kept in control, the project will have a much higher opportunity for success.

Work Breakdown Structure (WBS)

Step one in building a schedule is to begin with a Work Breakdown Structure (WBS) that allows decomposition of the scope of the project from major components to the smallest set of deliverables, called *work packages*. As a best practice, this process is completed before a true schedule is built. It

can be done using Project as long as ongoing "use" rules are defined and followed to keep the WBS components intact after the project is approved and baselined.

As mentioned earlier in this chapter, PMI has developed a standard for the WBS. It is a primary component of good project management practices because it forces the discipline of scope definition and control.

➥ *For detailed information of how to build a WBS,* ***see*** *Chapter 4, "Getting Started After the Business Initiative Is Approved,"* ***p. 71****.*

If the scope of your project is managed through a WBS, all the tasks and milestones will be created in support of specific work packages and can be rolled up through the structure for tracking progress using Earned Value Management techniques. This practice eliminates some of the common failure points in project management, such as scope creep and fuzzy requirements. All work is clearly linked to the production of a deliverable, and progress against that deliverable can be monitored.

Managerial Control

So many terms are used in the context of managerial control that a few definitions are in order. Hundreds of resources are available to provide detailed explanations; the purpose here is context only. The hope is that these simple descriptions will help the user's understanding when building a project schedule, as discussed in the following sections.

Phases and Gates

Many organizations have established processes for deciding what projects will be approved and for overseeing the projects after they have been launched. In some organizations, the processes are rigorous and robust; in others, the processes might be simple guidelines that have been put in place to help project managers. In either case, a defined set of standard phases and control points (often called *gates*) simplify the decisions that need to be made when running a project. In most cases, templates can be created that standardize the phases and the required control points for different types of projects.

Phases and gates can allow more management control of the project, as they break down the project into smaller components. This helps to keep executive and team focus aligned on the same set of activities. A change between phases is usually defined by some kind of transfer. In many cases, the transfer requires a formal review before the project is allowed to move into the next phase. It is not unusual, however, for phases to begin before the completion of the previous phase, especially when the risks are judged as acceptable. Each organization will make its own determination of the level of control required.

Building the phases and control points into templates is an excellent way to minimize the amount of work that needs to be done when building a new schedule. Many examples are already available in Project, and the organization can build additional ones as needed.

➥ *For additional information on building and using templates,* ***see*** *Chapter 18, "Managing Project Files Locally and in the Cloud," on* ***p. 549****.*

Methodologies

As organizations mature in the project management discipline, they often adopt more formal management control systems. These systems are typically described as methodologies that include processes, rules, standards, and methods for how work will be done. In this section, we identify a few of the ones used in specific industry segments. Each industry has its own set of methodologies, and this chapter does not attempt to identify all of them. The purpose here is to show how managing projects using Project can be included in the methodology to assist in the enforcement and usability of the tools.

Life Cycles

Like methodologies, project life cycles are unique to the industries and disciplines in which they are used. Although all projects have a beginning and an end, they vary greatly in how the work is accomplished. It is nearly impossible to define an ideal life cycle. Some companies and organizations use a single, standardized life cycle for every project, whereas others permit the project manager to choose the best life cycle for the project. In others, a variety of life cycles exists to accommodate different levels of complexity and different styles or types of work.

Regardless of the organization's choices regarding methodologies and life cycles, all organizations can use a scheduling tool to help with project execution. The key to success in every case is that the schedule must be focused on the deliverables to be produced rather than the process. The process must be set up to assist with producing deliverables.

The next section of this chapter provides several examples of methodologies and life cycles in the field of software development to illustrate how Project can be used to enable management of a wide variety of projects.

Using Microsoft Project with Methodologies and Life Cycles

Almost all organizations have at least a small number of technology projects underway, so software development is an excellent example of the wide variety of project-scheduling approaches available to organizations. The types of projects range from simple to complex, short to multiyear, and goal-oriented to open-ended research.

The following examples discuss the associated software development life cycle (SDLC) and how Project can be set up to support the life cycle. As you review the examples, you should also keep in mind that these projects should be planned and executed using the principles described in the previous sections on project standards (the PMBOK Guide and PRINCE2).

Although strict adherence to the standards is not required or necessary on every project, it is useful to remember that there are five major process groups to be managed on each project and that there are nine knowledge areas that should be considered throughout the project's life cycle.

Waterfall Development Process

Traditional software development is often described as a waterfall model because it is sequential in nature. The assumption with this model is that phases can be completed in order with little or no need to repeat the previous activities. Development is described as a waterfall, steadily falling down through traditional phases such as definition, preliminary design, detailed design, coding, testing, implementation, and transition to operations.

This method of development is used in many organizations today, especially those involved in multiyear programs. The phases can be lengthy and the work can be exacting. Although the name suggests that all work from one phase is completed before moving into the next phase, these types of projects are often set up with overlapping phases so that design can begin on certain deliverables as soon as the definition of the work for those deliverables is completed. In addition, there is typically some level of iterative development involved in almost all projects, but the term "waterfall" is still in common use today.

In this type of project, the tendency is to set up the project schedule in the same order as the major phase names. Instead, the project can be set up so that it is broken into logical work packages that can be monitored and measured separately.

Iterative Development

Iterative development provides a strong framework for planning purposes and also flexibility for successive iterations of software development. The Rational Unified Process (RUP) and the Dynamic Systems Development Methods are two frameworks for this type of project life cycle. RUP is not only a methodology for software engineering project management; it also has a set of software tools for using the specific methodology that is the Rational Unified Process. Figure 3.3 depicts the RUP workflow.

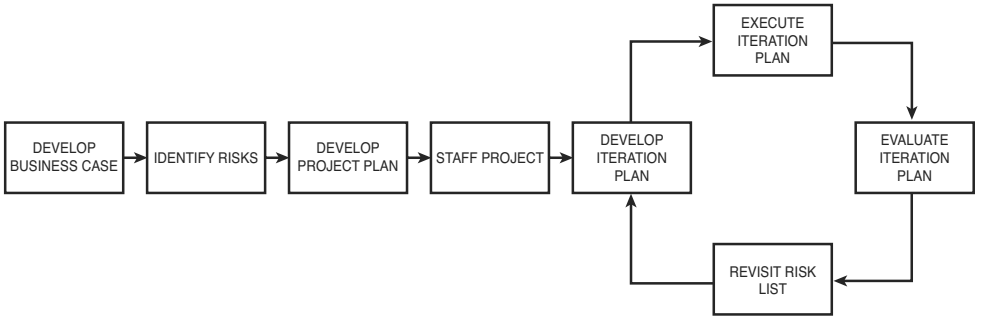

Figure 3.3 RUP workflow.

The goal for this type of software development life cycle (SDLC) is to allow the developers to learn through incremental development and the use of prototypes instead of trying to complete detailed requirements before the development work begins. Agile and XP can also be considered to be iterative methods.

Agile Development Process

Agile is a philosophy of project management that moves away from the classic project management methods and focuses less on planning and more on execution. In Agile, crucial decisions are made during the project execution phase, instead of the planning phase. As business and project environments become more fluid and dynamic, the amount of time for effective planning becomes less and less. This does not mean that planning is ignored; rather, the focus shifts to supporting decisions during project execution instead of finalizing all decisions during the planning stage.

Agile is not an all-or-nothing methodology either; it is possible to combine Agile with more classic project management ideas. Whereas classic project management is comprehensive and works in diverse situations, Agile can add various ideas for facing new and unique situations that can be found in creative, knowledge-based industries.

Here are some of the attributes of an Agile SDLC:

- Short development cycles are used to produce working software in weeks rather than months.
- Communication between the business users and the developers occurs daily.
- Documentation of working functionality is captured after the software is completed; there is limited documentation of the requirements or design.
- Timeboxing is used to force tough decisions early in the project.
- Changes to requirements are expected; they are a result of early working prototypes and are a goal of the process.
- The project manager for an Agile team is focused on ensuring excellent communication as the primary mechanism to maintain progress.

Agile development can be difficult for large organizations to embrace because it does not require a focus on formal planning of an entire project.

The major difference is that the primary measurement of progress is frequent delivery of small amounts of working software. With a focus on feature delivery, it can sometimes be difficult to understand the overall picture, so strong project management must provide this clarity.

caution

The use of Agile should not be used as an excuse to avoid planning or managing a budget. The approach is meant to provide a lighter and faster method to reach a goal, but the goal is still required.

In this type of environment, a project team can still use Project to support its goals. In an Agile environment, the tool is not used to develop a robust schedule with a beginning-to-end flow of tasks and resources. Its use in this case supports communication to management and ensures that changes are captured and the backlog of work is moved through each successive iteration of the project schedule.

In the following example, the project manager has established a budget summary task to provide rollup of budget for management purposes. Successive sets of work are defined in small iterations, while the overall timeframe and budget are obvious for all (see Figure 3.4). This approach enables the team to perform iterative planning while still meeting the business requirements of not exceeding a specific timeframe and budget.

By establishing a project schedule with an overall goal, the needs of the team and their management can be met. Refer to Figure 3.4 for an example of a short project that is expected to complete within a target effort of 340 hours. The work is not fully defined at the beginning so that the team has the flexibility to decide what work will happen in what order. Management is still able to see overall metrics of planned work, actual work, and the current estimate of work remaining.

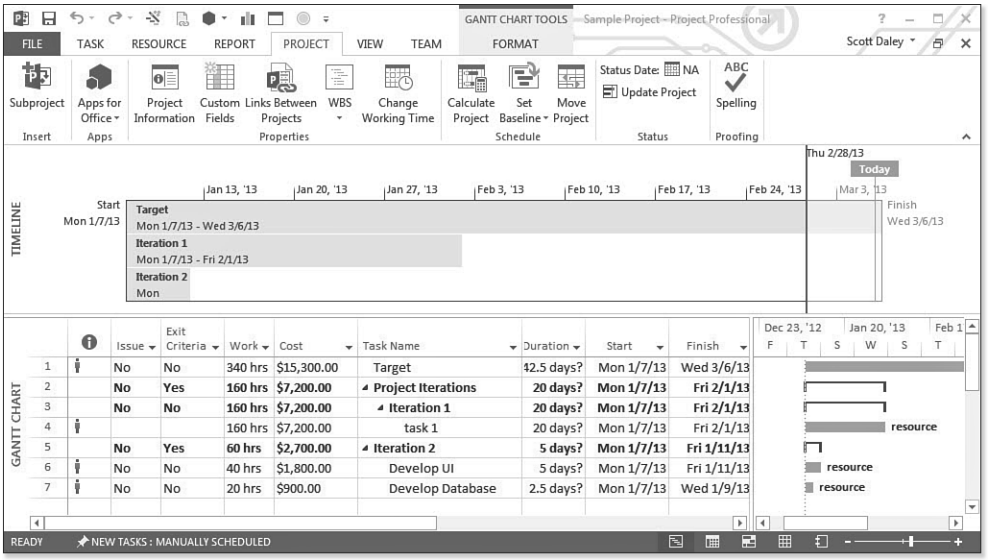

Figure 3.4 An Agile project showing overall budget, work, and timeframe with iterative development.

Agile is an extremely successful method of software development that is well suited to an environment with self-motivated teams, open communications, and leadership that is comfortable with a prototyping approach to work. It does not fit all projects, but when it works, it works well.

The schedule created in Project for this type of approach becomes a tool for communication, overall budget and time goals, and historical tracking purposes.

Extreme Programming

Extreme Programming, or XP, is another method within the Agile family that has become a simple and flexible way for developing software through the writing of tests. It is designed to be used by a group of two to ten programmers who are able to execute tests in a fraction of a day. It uses short cycles of phases, with early

note

For organizations that use the Project Server, this method enables them to use an Agile approach and yet have oversight of the entire project portfolio. Agile projects coexist with standard iterative projects in their Project Server environment; the projects have planned timeframes, resources, and budgets but are not required to have all the work scoped out at the beginning of the project.

and continuing feedback during each cycle. This flexibility enables it to respond to changing business demands through the implementation of functionality.

XP's use of automated tests, written by the programmers to scrutinize development, helps in early detection of defects and also enables the cycle of phases to evolve as the project continues. These automated tests depend both on the short-term instincts of programmers and also on the long-term interests of the project. XP also relies heavily on a system of oral communication, tests, and source code to help communicate the system structure and intent.

These processes allow for the day-to-day programming of a feature, and then moving on to testing, implementation, design, and integration, all packed into each cycle. The scheduling methods used in the preceding Agile example can again be adapted for XP.

Spiral Development Project

Spiral development was defined by Barry Boehm in 1985 and is often used in fairly large projects that take months to two years or more to complete. The initial focus might be on core functionality, and then the "bells and whistles" such as graphical user interfaces and reporting are added at a later time. This is sometimes considered to be another form of iterative development, but the structure of the plans and schedule focus on a robust core design in the early stages.

Research Project

A research project might be the most difficult type of project to tackle when it comes to constructing a project schedule. Often there is no clear goal in mind, and there might not even be an expectation of a specific end date or budget. On the other side, however, even research projects must be funded by someone, and they must have a working staff, so there is typically some expectation of a result. In most cases, there is also an expectation that the funding is used responsibly, so there must be a process in place to track how the money has been spent.

Project can once again be used to support this type of project as a tracking mechanism and a place to bring together the set of work that will be performed. The schedule will not require all the advanced features of critical path analysis, resource leveling, and predecessor/successor relationships, but it can be used as an easy method of historical support and a loose prediction of the work that is to be accomplished.

Accommodating Teaming Styles

High-performance teams, self-managed teams, and other nontraditional structures began to emerge more than 50 years ago in Great Britain and gained acceptance across the globe as several large corporations began to adopt the concepts. The general idea behind these teaming styles was to loosen managerial constraints in an effort to increase worker performance and make quantum leaps in accomplishment of organizational goals.

When framed correctly, the teams need little direction and excel in accomplishing the goals of their projects. If the dynamics are not understood, however, little is accomplished. From a project management perspective, Agile or XP projects can be a bit intimidating because the team dynamics can overwhelm the designated leader. In reality, successful self-managed teams are not leaderless. They

have simply figured out a mechanism to allow many people within the team to play a leadership role.

Even in a team where a project management role has not been defined, someone must take on the job of setting a direction to accomplish a goal. The goal might only be one week in the future, but the team must coalesce around that goal, and the person who makes that happen is a leader. If the project manager understands the dynamics of the team, he or she can use these dynamics to improve the team's focus and increase its performance. The PM must be comfortable with sharing decision making and needs to focus heavily on communication of information within the team and with the stakeholders of the project. Things change quickly in this environment, so communication of status becomes a driving force for the project.

Project is an excellent tool to aid the PM in communication. Two components need to be established to make this successful. The overall goal of the project needs to be clear to the team, and the boundaries of the project (overall timeframe, scope, resources, and budget) must be understood. If these components are established within the tool as a baseline, the remainder of the schedule can be flexible or rigid, as dictated by the project structure and the teaming style.

Consultants' Tips

Determine the Approach to Use in Managing Your Project

Project has a rich set of features that enable the project manager and team to track projects at a detailed level. It also has enough flexibility to allow high-level tracking without a demand for the detail. It can be and has been used to support all industries and all domains within those industries.

Because the software has so many capabilities, it must be well understood to be used correctly. The scheduling engine anticipates your needs and moves the dates or adjusts the amount of work that is to be accomplished based on the parameters that you set. Because it does this, project managers must have a clear understanding of the approach that they want to use on their projects before they begin entering tasks.

Use WBS as a First Step in Project Definition

Always start with a WBS to help you be clear on the goals of the project. Wait to add the task-level details until you are sure that you have decomposed the WBS to the work package level that is right for the type of project you are leading. Do not confuse the listing of activities with the completion of deliverables.

Use the 5×9 Checklist for Planning

Remember the 5×9 checklist and consider it when planning and executing each project. As you move through the phases of your project's life cycle, spend a moment to consider which of the five process groups is most dominant at the moment and which of the nine knowledge areas plays the most important part in the project's evolution.

This page intentionally left blank

INDEX

Symbols & Numerics

A

B

C

D

E

F

G

H

I

J-K

L

M

N

O

P

Q

R

S

T

W

X-Y-Z