

# CHAPTER 5

## PROJECT SCOPE MANAGEMENT

### LEARNING OBJECTIVES

**After reading this chapter, you will be able to:**

- List key reasons why good project scope management is important
- Describe the process of planning scope management
- Discuss methods for collecting and documenting requirements to meet stakeholder needs and expectations
- Explain the scope definition process and describe the contents of a project scope statement
- Discuss the process for creating a work breakdown structure using the analogy, top-down, bottom-up, and mind-mapping approaches
- Explain the importance of validating scope and how it relates to defining and controlling scope
- Given an information technology (IT) project situation, show how recommended approaches for controlling scope can improve the potential for project success
- Describe how software can assist in project scope management
- Discuss considerations for agile/adaptive environments

## OPENING CASE

Kim Nguyen was leading a meeting to create the work breakdown structure (WBS) for her company's IT upgrade project. This project was necessary because of several high-priority, Internet-based applications the company was developing. The IT upgrade project involved creating and implementing a plan to make all employees' IT assets meet new corporate standards within nine months. These standards specified the minimum requirements for each desktop or laptop computer, including the type of processor, amount of memory, hard disk size, type of network connection, security features, and software. Kim knew that to perform the upgrades, the project team would first have to create a detailed inventory of all the current hardware, networks, and software in the entire company of 2,000 employees.

Kim had worked with other stakeholders to develop a project charter and initial scope statement. The project charter included rough cost and schedule estimates for the project and signatures of key stakeholders; the initial scope statement provided a start in defining the hardware, software, and network requirements as well as other information related to the project scope. Kim called a meeting with her project team and other stakeholders to further define the scope of the project. She wanted to get everyone's ideas on what the project involved, who would do what, and how they could avoid scope creep. The company's new CEO, Walter Schmidt, was known for keeping a close eye on major projects. The company had started using a new project management information system that let everyone know the status of projects at a detailed and high level. Kim knew that a good WBS was the foundation for scope, time, and cost performance, but she had never led a team in creating one or allocating costs based on a WBS. Where should she begin?

## WHAT IS PROJECT SCOPE MANAGEMENT?

Many of the factors associated with project success, such as user involvement, clear business objectives, and optimized scope, are elements of project scope management.

A critically important and difficult aspect of project management is defining the scope of a project. **Scope** refers to *all* the work involved in creating the products of the project and the processes used to create them. Recall from Chapter 2 that a **deliverable** is a product created as part of a project. Deliverables can be product related, such as a piece of hardware or software, or process related, such as a planning document or meeting minutes. Project stakeholders must agree what the products of the project are and, to some extent, how they should be produced to define all of the deliverables.

**Project scope management** includes the processes involved in defining and controlling what work is or is not included in a project. It ensures that the project team and stakeholders have the same understanding of what products the project will produce and what processes the project team will use to produce them. Six main processes are involved in project scope management:

1. *Planning scope management* involves determining how the project's scope and requirements will be managed.
2. *Collecting requirements* involves defining and documenting the features and functions of the products as well as the processes used for creating them.

3. *Defining scope* involves reviewing the scope management plan, project charter, requirements documents, and organizational process assets to create a scope statement, adding more information as requirements are developed and change requests are approved.
4. *Creating the WBS* involves subdividing the major project deliverables into smaller, more manageable components.
5. *Validating scope* involves formalizing acceptance of the project deliverables. Key project stakeholders, such as the customer and sponsor for the project, inspect and then formally accept the deliverables during this process. If the deliverables are not acceptable, the customer or sponsor usually requests changes.
6. *Controlling scope* involves controlling changes to project scope throughout the life of the project—a challenge on many IT projects. Scope changes often influence the team's ability to meet project time and cost goals, so project managers must carefully weigh the costs and benefits of scope changes.

Figure 5-1 summarizes the inputs, tools and techniques, and outputs of project scope management.

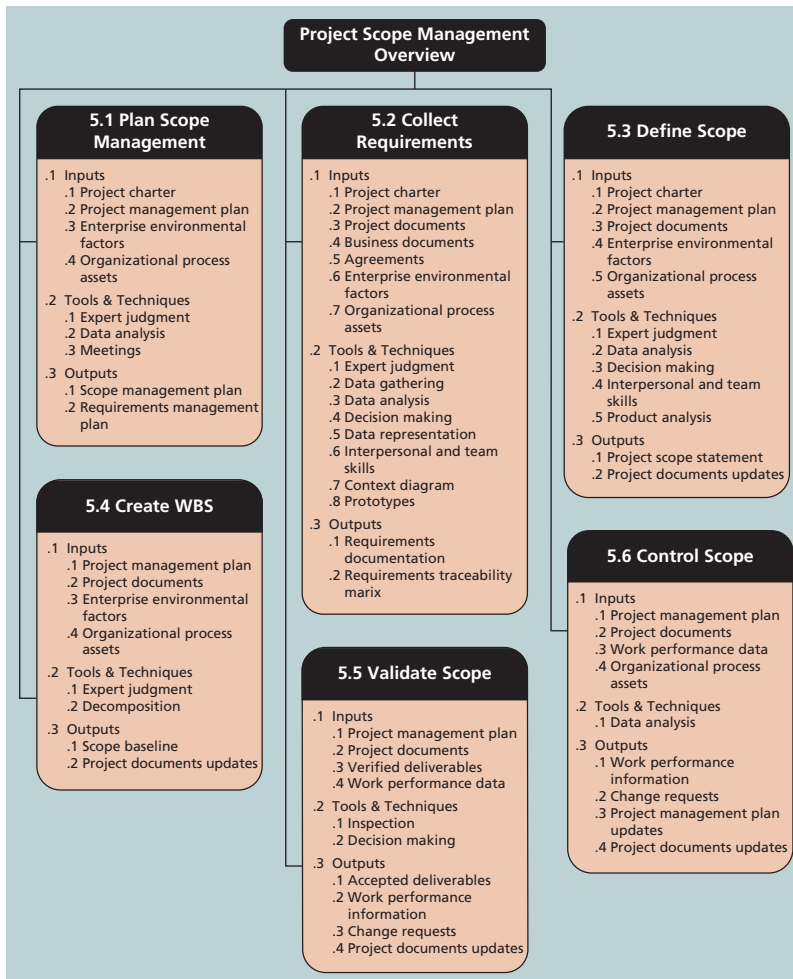
## PLANNING SCOPE MANAGEMENT

---

The first step in project scope management is planning how the scope will be managed throughout the life of the project. After reviewing the project management plan, project charter, enterprise environmental factors, and organizational process assets, the project team uses expert judgment, data analysis, and meetings to develop two important outputs: the scope management plan and the requirements management plan.

The scope management plan is a subsidiary part of the project management plan, as described in Chapter 4, Project Integration Management. It can be informal and broad or formal and detailed, based on the needs of the project. In fact, small projects may not need a written scope management plan, but large projects or highly technical projects often benefit from one. In general, a scope management plan includes the following information:

- *How to prepare a detailed project scope statement*: For example, are there templates or guidelines to follow? How much detail is needed to describe each deliverable?
- *How to create a WBS*: It is often difficult to create a good WBS. This section of the scope management plan would provide suggestions, samples, and resources for creating a WBS.
- *How to maintain and approve the WBS*: The initial WBS often changes, and project team members disagree on what should be included. The scope management plan describes guidelines for maintaining the WBS and getting approval for it.
- *How to obtain formal acceptance of the completed project deliverables*: It is extremely important to understand the process for obtaining formal acceptance of completed deliverables, especially for projects in which payments are based on formal acceptance.



Source: *PMBOK® Guide – Sixth Edition*. Project Management Institute, Inc. (2017). Copyright and all rights reserved. Material from this publication has been reproduced with permission of PMI.

**FIGURE 5-1** Project scope management overview

- *How to control requests for changes to the project scope:* This process is related to performing integrated change control, as described in Chapter 4. Organizations often have guidelines for submitting, evaluating, and approving changes to scope, and this section of the scope management plan would specify how to handle change requests for the project.

Another important output of planning scope management is the requirements management plan. Before you learn about the contents of this document, it is important to understand what requirements are. The 1990 IEEE Standard Glossary of Software Engineering Terminology defines a requirement as follows:

1. A condition or capability needed by a user to solve a problem or achieve an objective.
2. A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed document.
3. A documented representation of a condition or capability as in 1 or 2.<sup>1</sup>

The *PMBOK® Guide – Sixth Edition*, defines a **requirement** as “a condition or capability that is necessary to be present in a product, service, or result to satisfy a business need.” It further explains that requirements “include the quantified and documented needs and expectations of the sponsor, customer, and other stakeholders. These requirements need to be elicited, analyzed, and recorded in enough detail to be included in the scope baseline and be measured once project execution begins.”<sup>2</sup>

For example, the chapter’s opening case describes a project for upgrading IT assets to meet corporate standards. These standards specify the minimum requirements for each laptop, such as the type of processor, amount of memory, and hard disk size. The documented requirements for this project, therefore, might state that all laptops include a specific type of processor, a minimum amount of memory, and a minimum hard drive size.

For software projects, it is helpful to divide requirements development into the software engineering steps called *elicitation*, *analysis*, *specification*, and *validation*. These steps include all the activities involved in gathering, evaluating, and documenting requirements for a software or software-containing product. It is also important to use an iterative approach to defining requirements because they are often unclear early in a project.

The **requirements management plan** documents how project requirements will be analyzed, documented, and managed. A requirements management plan can include the following information:

- How to plan, track, and report requirements activities
- How to perform configuration management activities
- How to prioritize requirements
- How to use product metrics
- How to trace and capture attributes of requirements



## WHAT WENT RIGHT?

Several studies cite how difficult it is to manage requirements. Finding qualified people—business analysts—to do the job is equally difficult. The U.S. Bureau of Labor Statistics has projected the number of jobs for business analysts to increase 19 percent by 2022.<sup>3</sup> A PMI survey found that only 49 percent of respondents had the resources in place to do requirements management properly and 53 percent failed to use a formal process to validate requirements.<sup>4</sup>

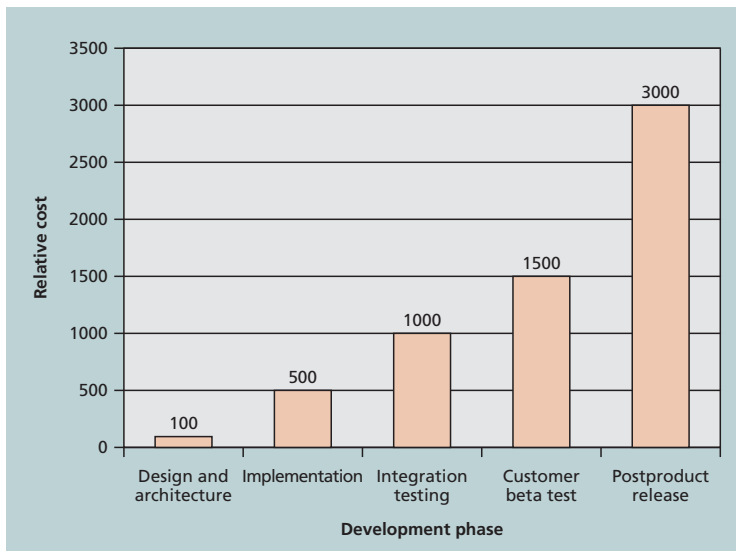
(continued)

Fortunately, several organizations have recognized this need and have developed training and certification programs for business analysts.

- The International Institute of Business Analysis (IIBA®) issues both the Certified Business Analysis Professional (CBAP®) and Certification of Competency in Business Analysis (CCBA®) certifications. This organization had over 29,000 members in 2018 and published the third edition of the *Guide to Business Analysis Body of Knowledge (BABOK® Guide)*.
- The International Requirements Engineering Board (IREB) provides certification as a Certified Professional for Requirements Engineering (CPRE). Over 37,000 people in 73 countries held this certification in 2018.
- PMI began offering a new certification as a PMI Professional in Business Analysis (PMI-PBA®) in 2014.
- Several colleges and universities are offering majors and minors in business or data analytics at the undergraduate and graduate levels.

## COLLECTING REQUIREMENTS

The second step in project scope management is often the most difficult: collecting requirements. A major consequence of not defining requirements well is rework, which can consume up to half of project costs, especially for software development projects. As illustrated in Figure 5-2, it costs much more (up to 30 times more) to correct a software



Source: IBM Software Group, “Minimizing code defects to improve software quality and lower development costs,” Rational Software (October 2008).

**FIGURE 5-2** Relative cost to correct a software defect

defect in later development phases than to fix it in the requirements phase. Everyone can cite examples in all types of industries of how important it is to understand requirements as early as possible. For example, if you are designing a house, it is much cheaper to decide where windows and walls will be on paper or a computer screen than after the entire house is framed. New processes and technologies are making it easier to define and implement requirements, but it is still one of the most challenging aspects of project scope management.



## BEST PRACTICE

Google continues to be among the most admired companies in the world, well-known for being a great place to work and for being innovative in developing new products. James Whittaker, former engineering director at Google, responsible for testing Chrome, maps, and Google Web apps, wrote a book with co-authors Jason Arbon and Jeff Carollo (also former Google employees) called *How Google Tests Software*. Whittaker also worked for Microsoft and as a professor and is one of the best-known names in testing software.

In *How Google Tests Software*, the authors explain that Google used to be like other large companies: testing was not part of the mainstream, and people who did it were underappreciated and overworked. It took Google a long time to develop the people, processes, and technologies it uses today to develop software. A key part of success at Google was changing the culture. Google has fewer dedicated testers than most of its competitors have on a single product team. How is that possible? The authors explain that quality rests on the shoulders of those writing the code. The employees at Google do things the best, fastest way possible for them, knowing that coders are responsible for the quality of their own work. They don't rely on testers to ensure quality.

It is also interesting to note that Google does not believe in fads or buzzwords. In an interview, InfoQ asked the authors how Google's approach fits into the wider Agile community. Their response: "Google doesn't try to be part of the Agile community. We don't use the terminology of Scrums or bother with Scrum masters and the like. We have crafted our own process of moving fast. It's a very Agile process that doesn't get bogged down with someone else's idea of what it means to be Agile. When you have to stop and define what it means to be Agile and argue what flavor of Agile you are, you just stopped being Agile."<sup>5</sup>

Part of the difficulty is that people often do not have a good process for collecting and documenting project requirements.

There are several ways to collect requirements. Interviewing stakeholders one on one is often very effective, although it can be expensive and time consuming. Holding focus groups and facilitated workshops, and using group creativity and

decision-making techniques to collect requirements are normally faster and less expensive than one-on-one interviews. Questionnaires and surveys can be efficient ways to collect requirements as long as key stakeholders provide honest and thorough information. Observation can also be a good technique for collecting requirements, especially for projects that involve improving work processes and procedures. For software development projects, prototyping and document analysis are common techniques for collecting requirements, as are context diagrams, which help to clarify the interfaces and boundaries of a project or process. On agile software development projects, the product owner creates the prioritized product backlog for each sprint, as shown in Chapter 3. **Benchmarking**, or generating ideas by comparing specific project practices or product characteristics to those of other projects or products inside or outside the performing organization, can also be used to collect requirements.

Even though there are many ways to collect requirements, people who work on software projects in particular have considerable difficulty defining and managing requirements. One study revealed some interesting statistics:

- Eighty-eight percent of the software projects involved enhancing existing products instead of creating new ones.
- Eighty-six percent of respondents said that customer satisfaction was the most important metric for measuring the success of development projects, 82 percent said that feedback from customers and partners was the main source of product ideas and requirements, and 73 percent said that the most important challenge for their teams was gaining a clear understanding of what customers wanted, followed by documenting and managing requirements.
- Seventy-five percent of respondents were managing projects with at least 100 requirements; 20 percent were managing projects with over 1,000 requirements.
- Seventy percent of respondents spent at least 10 percent of their time managing changes to requirements; 30 percent spent more than 25 percent of their time on such changes.
- The majority of software development teams used a hybrid methodology, 26 percent used waterfall or modified waterfall techniques, and 19 percent used agile techniques.
- Eighty-three percent of software development teams still use Microsoft Office applications such as Word and Excel as their main tools to communicate requirements.
- The respondents listed “requirements collaboration and management software” and “requirements modeling and visualization” as the top two software tools on their wish list, followed by test management and project management.<sup>6</sup>

The project’s size, complexity, importance, and other factors affect how much effort is spent on collecting requirements. For example, a team working on a project to upgrade the entire corporate accounting system for a multibillion-dollar company with



**TABLE 5-1** Sample entry in a requirements traceability matrix

Requirement No.	Name	Category	Source	Status
R32	Laptop memory	Hardware	Project charter and corporate laptop specifications	Complete. Laptops ordered meet memory requirement.

more than 50 locations should spend a fair amount of time collecting requirements. On the other hand, a project to upgrade the hardware and software for a small accounting firm with only five employees would need a much smaller effort. In any case, it is important for a project team to decide how it will collect and manage requirements. It is crucial to gather inputs from key stakeholders and align the scope with business strategy, as described in Chapter 4.

Just as a project team can collect requirements in several ways, there are several ways to document the requirements. Project teams should first review the project charter because it includes high-level requirements for the project, and they should refer to the scope and requirements management plans. They should also review the stakeholder register and stakeholder management plan to ensure that all key stakeholders have a say in determining requirements. The format for documenting stakeholder requirements can range from a listing of all requirements on a single piece of paper to a room full of notebooks. People who have worked on complex projects, such as building a new airplane, know that the requirements documentation for a plane can weigh more than the plane itself! Requirements documents are often generated by software and include text, images, diagrams, videos, and other media. Requirements are often broken down into different categories such as functional requirements, service requirements, performance requirements, quality requirements, and training requirements.

In addition to preparing requirements documentation as an output of collecting requirements, project teams often create a requirements traceability matrix. A **requirements traceability matrix (RTM)** is a table that lists requirements, their various attributes, and the status of the requirements to ensure that all are addressed. Table 5-1 provides an example of an RTM entry for the IT upgrade project described in the chapter's opening case. An RTM can have many variations. For example, software requirements are often documented in an RTM that cross-references each requirement with related ones and lists specific tests to verify that they are met. Remember that the main purpose of an RTM is to maintain the linkage from the source of each requirement through its decomposition to implementation and validation. Search the Internet for more detailed examples of an RTM.

## DEFINING SCOPE

The next step in project scope management is to provide a detailed definition of the work required for the project. Good scope definition is very important to project success because it helps improve the accuracy of time, cost, and resource estimates, it defines a

baseline for performance measurement and project control, and it aids in communicating clear work responsibilities. The main tools and techniques used in defining scope include expert judgment, data analysis, decision making, interpersonal and team skills, and product analysis. The main outputs of scope definition are the project scope statement and project documents updates.

Table 5-2 shows the project charter for the IT upgrade project described in the opening case. Notice how information from the project charter provides a basis for further defining the project scope. The charter describes the high-level scope, time, and cost goals for the project objectives and success criteria, a general approach to accomplishing the project's goals, and the main roles and responsibilities of important project stakeholders.

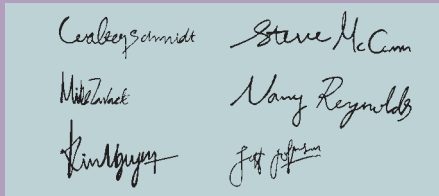
Although contents vary, **project scope statements** should include at least a product scope description, product user acceptance criteria, and detailed information on all project deliverables. It is also helpful to document other scope-related information, such as the project boundaries, constraints, and assumptions. The project scope

**TABLE 5-2** Sample project charter

<b>Project Title:</b> Information Technology (IT) Upgrade Project	
<b>Project Start Date:</b> March 4	<b>Projected Finish Date:</b> December 4
<b>Key Schedule Milestones:</b> <ul style="list-style-type: none"><li>• Inventory update completed April 15</li><li>• Hardware and software acquired August 1</li><li>• Installation completed October 1</li><li>• Testing completed November 15</li></ul>	
<b>Budget Information:</b> Budgeted \$1,000,000 for hardware and software costs and \$500,000 for labor costs.	
<b>Project Manager:</b> Kim Nguyen, (310) 555-2784, knguyen@course.com	
<b>Project Objectives:</b> Upgrade hardware and software for all employees (approximately 2,000) within nine months based on new corporate standards. See attached sheet describing the new standards. Upgrades may affect servers as well as associated network hardware and software.	
<b>Main Project Success Criteria:</b> The hardware, software, and network upgrades must meet all written specifications, be thoroughly tested, and be completed in nine months. Employee work disruptions will be minimal.	
<b>Approach:</b> <ul style="list-style-type: none"><li>• Update the IT inventory database to determine upgrade needs.</li><li>• Develop detailed cost estimate for project and report to CIO.</li><li>• Issue a request for quote to obtain hardware and software.</li><li>• Use internal staff as much as possible for planning, analysis, and installation.</li></ul>	

(continued)

**TABLE 5-2** Sample project charter (*continued*)

ROLES AND RESPONSIBILITIES		
Name	Role	Responsibility
Walter Schmidt	CEO	Project sponsor, monitor project
Mike Zwack	CIO	Monitor project, provide staff
Kim Nguyen	Project Manager	Plan and execute project
Jeff Johnson	Director of IT Operations	Mentor Kim
Nancy Reynolds	VP, Human Resources	Provide staff, issue memo to all employees about project
Steve McCann	Director of Purchasing	Assist in purchasing hardware and software
Sign-off: (Signatures of all the above stakeholders)		
		
<b>Comments:</b> (Handwritten or typed comments from above stakeholders, if applicable) “This project must be done within 10 months at the absolute latest.”—Mike Zwack, CIO “We are assuming that adequate staff will be available and committed to supporting this project. Some work must be done after hours to avoid work disruptions, and overtime will be provided.”—Jeff Johnson and Kim Nguyen, IT department		

statement should also reference supporting documents, such as product specifications that will affect what products are created or purchased, or corporate policies, which might affect how products or services are produced. Many IT projects require detailed functional and design specifications for developing software, which also should be referenced in the detailed scope statement.

As time progresses, the scope of a project should become more clear and specific. For example, the project charter for the IT upgrade project in Table 5-2 includes a short statement about the servers and other computers and software that the project may affect. Table 5-3 provides an example of how the scope becomes progressively more detailed in scope statements labeled Version 1 and Version 2.

Notice in Table 5-3 that the project scope statements often refer to related documents, which can be product specifications, product brochures, or other plans. As more information becomes available and decisions are made related to project scope,

**TABLE 5-3** Further defining project scope

<b>Project Charter:</b> Upgrades may affect servers . . . (listed under Project Objectives)
<b>Project Scope Statement, Version 1:</b> Servers: If additional servers are required to support this project, they must be compatible with existing servers. If it is more economical to enhance existing servers, a detailed description of enhancements must be submitted to the CIO for approval. See current server specifications provided in Attachment 6. The CEO must approve a detailed plan describing the servers and their location at least two weeks before installation.
<b>Project Scope Statement, Version 2:</b> Servers: This project will require purchasing 10 new servers to support Web, network, database, application, and printing functions. Virtualization will be used to maximize efficiency. Detailed descriptions of the servers are provided in a product brochure in Attachment 8, along with a plan describing where they will be located.

such as specific products that will be purchased or changes that have been approved, the project team should update the project scope statement. The team might name different iterations of the scope statement Version 1, Version 2, and so on. These updates may also require changes to other project documents. For example, if the company must purchase servers from a supplier it has never worked with before, the procurement management plan should include information on working with that new supplier.

An up-to-date project scope statement is an important document for developing and confirming a common understanding of the project scope. It describes in detail the work to be accomplished on the project and is an important tool for ensuring customer satisfaction and preventing scope creep, as described later in this chapter.

Recall from Chapter 1 the importance of addressing the triple constraint of project management—meeting scope, time, and cost goals for a project. Time and cost goals are normally straightforward. For example, the time goal for the IT upgrade project is nine months, and the cost goal is \$1.5 million. It is much more difficult to describe, agree upon, and meet the scope goal of many projects.



**MEDIA SNAPSHOT**

Inaccurate requirements gathering continues to be one of the main causes of project failure. Organizations lack the resources and maturity in requirements management, which comes at a price. The authors of the 2014 PMI in-depth report on requirements management state that “For every dollar spent on projects and programs, 5.1 percent is wasted due to poor requirements management. Put in more striking terms, this amounts

(continued)

to US\$51 million wasted for every US\$1 billion spent. That's a lot of potential value falling through the cracks in the project-driven world."<sup>7</sup>

The report provides the following survey results:

- Forty-seven percent of organizations report that inaccurate requirements management is the primary cause of projects not meeting their original goals and business objectives.
- Over half of organizations report not having adequate resources to perform requirements management well.
- Only 24 percent of organizations report doing well in recognizing and developing skills needed for effective management of requirements.
- Only 46 percent of organizations report using a formal process to ensure unbiased requirements validation for projects.

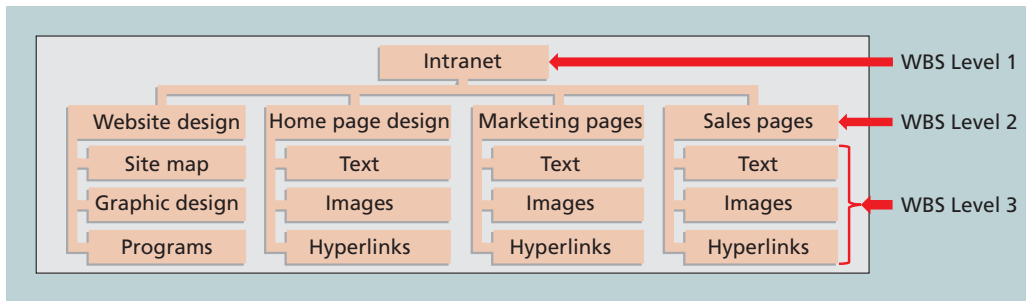
The report concludes by suggesting that organizations need to develop people, processes, and culture to improve requirements management.

## CREATING THE WORK BREAKDOWN STRUCTURE

After collecting requirements and defining scope, the next step in project scope management is to create a work breakdown structure. A **work breakdown structure (WBS)** is a deliverable-oriented grouping of the work involved in a project that defines its total scope. Because most projects involve many people and many different deliverables, it is important to organize and divide the work into logical parts based on how the work will be performed. The WBS is a foundation document in project management because it provides the basis for planning and managing project schedules, costs, resources, and changes. Because the WBS defines the total scope of the project, some project management experts believe that work should not be done on a project if it is not included in the WBS. Therefore, it is crucial to develop a complete WBS.

The project management plan, project documents, enterprise environmental factors, and organizational process assets are the primary inputs for creating a WBS. In addition to expert judgment, the main tool or technique is **decomposition**—that is, subdividing project deliverables into smaller pieces. The outputs of the process of creating the WBS are the scope baseline and project documents updates. The **scope baseline** includes the approved project scope statement and its associated WBS and WBS dictionary.

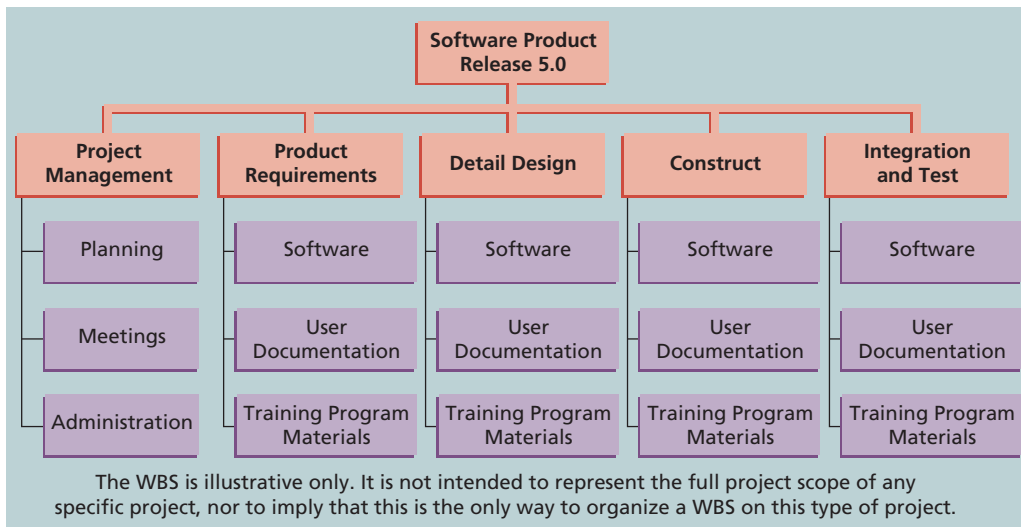
What does a WBS look like? A WBS is often depicted as a task-oriented tree of activities, similar to an organizational chart. Many people like to create a WBS in chart form first to help them visualize the whole project and all of its main parts. Figure 5-3 shows a WBS for an intranet project. The top box labeled Intranet represents the entire project, and it is called Level 1 of the WBS. In this example, the main products to be delivered are shown in Level 2, called website design, home page design, marketing pages, and sales pages. Below them are Level 3 items, such as site map, graphic design, and so on.



**FIGURE 5-3** Sample intranet WBS organized by product

Figure 5-4 shows a software product release project created by PMI. Notice that instead of showing product deliverables in Level 2, it is organized around deliverables for each project phase (product requirements, detail design, construct, and integration and test) as well as another Level 2 item for project management. Table 5-4 shows this same WBS in a tabular or list format. This level numbering is based on the Project Management Institute's *Practice Standard for Work Breakdown Structures, Second Edition* (2006).

Many documents, such as contracts, use the tabular format, but the numbering scheme may vary. Project management software also uses the tabular format. The WBS provides the basis for creating the activities that are entered into the Task Name column in Microsoft Project, and the hierarchy or level of tasks is shown by indenting



Source: *PMBOK® Guide – Sixth Edition*. Project Management Institute, Inc. (2017). Copyright and all rights reserved. Material from this publication has been reproduced with permission of PMI.

**FIGURE 5-4** Sample intranet WBS organized by phase in chart and tabular form

**TABLE 5-4** Tabular form of WBS

1.0	Software Product Release 5.0
1.1	Project Management
1.1.1	Planning
1.1.2	Meetings
1.1.3	Administration
1.2	Product Requirements
1.2.1	Software
1.2.2	User Documentation
1.2.3	Training Program Materials
1.3	Detail Design
1.3.1	Software
1.3.2	User Documentation
1.3.3	Training Program Materials
1.4	Construct
1.4.1	Software
1.4.2	User Documentation
1.4.3	Training Program Materials
1.5	Integration and Test
1.5.1	Software
1.5.2	User Documentation
1.5.3	Training Program Materials

and numbering tasks within the software. Be sure to check with your organization to see which numbering scheme it prefers to use for WBSs. To avoid confusion, it is important to decide on a numbering scheme and then use it when referring to WBS items.

Also determine how you will name WBS items. Some organizations use only nouns to focus on describing deliverables, as PMI recommends. For example, instead of using the words “define requirements” on a WBS, use “requirements definition.” Activities on a schedule below requirements definition would include verb phrases, as discussed in Chapter 6, Project Schedule Management. Remember that the main purpose of the WBS is to define all of the work required to complete a project, so focus on that while following your organization’s structural guidelines.

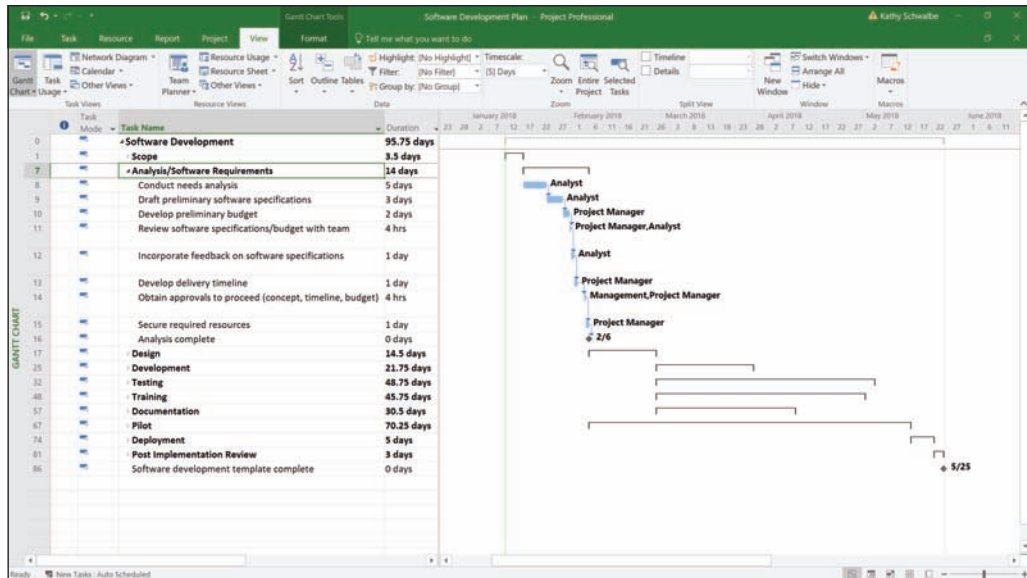
A **work package** is a task at the lowest level of the WBS. A work package also represents the level of work that the project manager monitors and controls. You can think of work packages in terms of accountability and reporting. If a project has a relatively short time frame and requires weekly progress reports, a work package might represent work completed in one week or less. If a project has a very long time frame and requires quarterly progress reports, a work package might represent work completed in one month or more. A work package might also be the procurement of a specific product or products, such as an item or items purchased from an outside source. A work package should be defined at the proper level so the project manager can clearly establish an estimate of the effort needed to complete it, estimate the cost of all required resources, and evaluate the quality of the results when the work package is finished.

When using project management software, estimates of work time should be entered only at the work package level. The rest of the WBS items are just groupings or summary tasks for the work packages. The software automatically calculates duration estimates for various WBS levels based on data entered for each work package and the WBS hierarchy.

Figure 5-5 shows a template file from Microsoft Project 2016 for a Software Development project. The Level 2 WBS categories include scope, analysis/software requirements, design, development, testing training, documentation, pilot, deployment, and post implementation review. Activities are shown for the analysis/software requirements. Notice that the WBS is the basis for creating the Gantt chart, as described in more detail in Chapter 6, Project Schedule Management. You will learn more about Gantt charts in Chapter 6, Project schedule management.

The example WBSs shown here are simplified so that they are somewhat easy to understand and construct. *Nevertheless, it is very difficult to create a good WBS.* To create a good WBS, you must understand the project and its scope and incorporate the needs and knowledge of the stakeholders. The project manager and the project team must decide as a group how to organize the work and how to include many levels in the WBS.

While many project managers have found that they should focus on doing the top levels well before becoming bogged down in more detailed levels, it is also true that more accurate estimates of scope, time, and cost are obtained when the project is defined appropriately and in sufficient detail. Operating at too high a level increases project risk. The act of defining the WBS is meant to offset that risk by considering the project's details in advance of their execution.

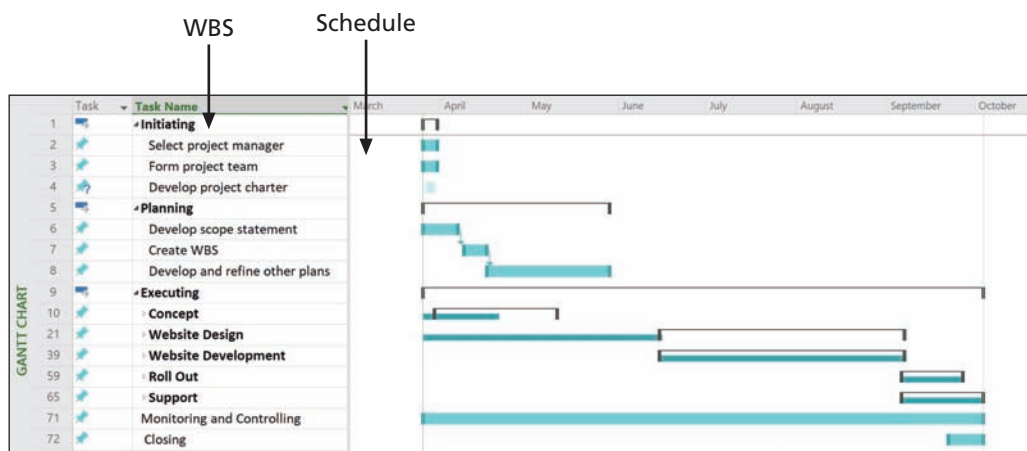


**FIGURE 5-5** Software development project template from Microsoft Project 2016



Many people confuse tasks on a WBS with specifications. Tasks on a WBS represent work that needs to be done to complete the project. For example, if you are creating a WBS to redesign a kitchen, you might have Level 2 categories called design, purchasing, flooring, walls, cabinets, and appliances. Under flooring, you might have tasks to remove the old flooring, install the new flooring, and install the trim. You would not have tasks like “12 ft. by 14 ft. of light oak” or “flooring must be durable”; these are specifications.

Another concern when creating a WBS is how to organize it to provide the basis for the project schedule. You should focus on what work needs to be done and how it will be done, not when it will be done. In other words, the tasks do not have to be developed as a sequential list of steps. If you want some time-based flow for the work, you can create a WBS using the project management process groups of initiating, planning, executing, monitoring and controlling, and closing as Level 2 in the WBS. By doing this, not only does the project team follow good project management practice, the WBS tasks can also be mapped more easily against time. For example, Figure 5-6 shows a WBS and Gantt chart for a website project, organized by the five project management process groups. Activities under initiating include selecting a project manager, forming the project team, and developing the project charter. Activities under planning include developing a scope statement, creating a WBS, and developing and refining other plans, which would be broken down in more detail for a real project. Concept, website design, website development, and rollout are WBS Level 3 items under executing. The executing tasks vary the most from project to project, but many of the activities under the other project management process groups would be similar for all projects. If you do not use the project management process groups in the WBS, you can have a Level 2 category called “project management” to make sure you account for all activities related to managing the project, as was done in Figure 5-4 from PMI. Remember that all work should be included in the WBS, including project management.



**FIGURE 5-6** Website project Gantt chart organized by project management process groups

**TABLE 5-5** Executing tasks for JWD Consulting's WBS

3.0	Executing
3.1	Survey
3.2	User inputs
3.3	Intranet site content
3.3.1	Templates and tools
3.3.2	Articles
3.3.3	Links
3.3.4	Ask the Expert
3.3.5	User Requests feature
3.4	Intranet site design
3.5	Intranet site construction
3.6	Intranet site testing
3.7	Intranet site roll-out
3.8	Site rollout
3.9	Intranet site roll-out

Some project teams like to list every deliverable they need to produce and then use those as the basis for creating all or part of their WBS. In Chapter 3, JWD Consulting used the project management process groups for the Level 2 items in its WBS for the project management intranet site project. Then in breaking down the executing section, the project team focused on the product deliverables it had to produce. Table 5-5 shows the categories the team used for that part of the WBS. Recall that the scope statement should list and describe all of the deliverables required for the project. It is very important to ensure consistency between the project charter, scope statement, WBS, and Gantt chart to define the scope of the project accurately.

It is also very important to involve the entire project team and the customer in creating and reviewing the WBS. *People who will do the work should help to plan the work* by creating the WBS. Having group meetings to develop a WBS helps everyone understand *what* work must be done for the entire project and *how* it should be done, given the people involved. It also helps to identify where coordination between different work packages will be required.

## Approaches to Developing Work Breakdown Structures

You can use several approaches to develop a WBS:

- Using guidelines
- The analogy approach
- The top-down approach
- The bottom-up approach
- The mind-mapping approach

## Using Guidelines

If guidelines exist for developing a WBS, it is very important to follow them. Some organizations—the U.S. Department of Defense (DOD), for example—prescribe the form and content for WBSs for particular projects. Many DOD projects require contractors to prepare their proposals based on the DOD-provided WBS. These proposals must include cost estimates for each task in the WBS at a detailed and summary level. The cost for the entire project must be calculated by summing the costs of all of the lower-level WBS tasks. When DOD personnel evaluate cost proposals, they must compare the contractors' costs with the DOD's estimates. A large variation in costs for a certain WBS task often indicates confusion as to what work must be done.

Consider a large automation project for the U.S. Air Force. In the mid-1980s, the Air Force developed a request for proposals for the Local On-Line Network System (LONS) to automate 15 Air Force Systems Command bases. This \$250 million project involved providing the hardware and developing software for sharing documents such as contracts, specifications, and requests for proposals. The Air Force proposal guidelines included a WBS that contractors were required to follow in preparing their cost proposals. Level 2 WBS items included hardware, software development, training, and project management. The hardware item was composed of several Level 3 items, such as servers, workstations, printers, and network hardware. Air Force personnel reviewed the contractors' cost proposals against their internal cost estimate, which was also based on this WBS. Having a prescribed WBS helped contractors to prepare their cost proposals and helped the Air Force to evaluate them.

Many organizations provide guidelines and templates for developing WBSs, as well as examples of WBSs from past projects. Microsoft Project 2016 comes with several templates, and more are available on Microsoft's website and other sites. At the request of many of its members, PMI developed a WBS *Practice Standard* to provide guidance for developing and applying the WBS to project management. The *Practice Standard* includes sample WBSs for a wide variety of projects in various industries, including projects for Web design, telecom, service industry outsourcing, and software implementation.

Project managers and their teams should review appropriate information to develop their unique project WBSs more efficiently. For example, Kim Nguyen and key team members from the opening case should review their company's WBS guidelines, templates, and other related information before and during the team meetings to create their WBS.

## The Analogy Approach

Another method for constructing a WBS is the analogy approach. In the **analogy approach**, you use a similar project's WBS as a starting point. For example, Kim Nguyen from the opening case might learn that one of her organization's suppliers did a similar IT upgrade project last year. She could ask them to share their WBS for that project to provide a starting point for her own project.

McDonnell Aircraft Company, now part of Boeing, provides an example of using an analogy approach when creating WBSs. McDonnell Aircraft Company designed and manufactured several different fighter aircraft. When creating a WBS for a new aircraft

design, it started by using 74 predefined subsystems for building fighter aircraft based on past experience. There was a Level 2 WBS item for the airframe that was composed of Level 3 items such as a forward fuselage, center fuselage, aft fuselage, and wings. This generic, product-oriented WBS provided a starting point for defining the scope of new aircraft projects and developing cost estimates for new aircraft designs.

Some organizations keep a repository of WBSs and other project documentation on file to assist people working on projects. Project 2016 and many other software tools include sample files to assist users in creating a WBS and Gantt chart. Viewing examples of WBSs from similar projects allows you to understand different ways to create a WBS.

### The Top-Down and Bottom-Up Approaches

Two other methods of creating WBSs are the top-down and bottom-up approaches. Most project managers consider the top-down approach of WBS construction to be conventional.

To use the **top-down approach**, start with the largest items of the project and break them into subordinate items. This process involves refining the work into greater and greater levels of detail. After finishing the process, all resources should be assigned at the work package level. The top-down approach is best suited to project managers who have vast technical insight and a big-picture perspective.

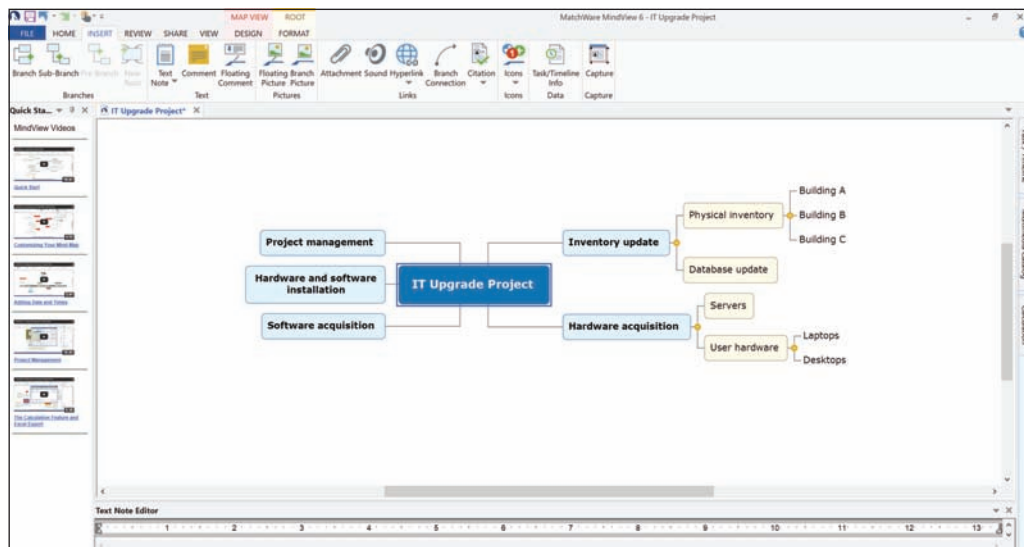
In the **bottom-up approach**, team members first identify as many specific tasks related to the project as possible. They then aggregate the specific tasks and organize them into summary activities, or higher levels in the WBS. For example, a group of people might be responsible for creating a WBS to develop an e-commerce application. Instead of looking for guidelines on how to create a WBS or viewing WBSs from similar projects, they could begin by listing detailed tasks they think they would need to perform in order to create the application. After listing these detailed tasks, they would group the tasks into categories. Then they would group these categories into higher-level categories. Some people have found that writing all possible tasks as notes and then placing them on a wall helps the team see all the work required for the project and develop logical groupings for performing the work. For example, a business analyst on the project team might know that it had to define user requirements and content requirements for the e-commerce application. These tasks might be part of the requirements documents the team would have to create as one of the project deliverables. A hardware specialist might know that the team had to define system requirements and server requirements, which would also be part of a requirements document. As a group, they might decide to put all four of these tasks under a higher-level item called “requirements definition” that would result in the delivery of a requirements document. Later, they might realize that requirements definition should fall under a broader category of concept design for the e-commerce application, along with other groups of tasks related to the concept design. The bottom-up approach can be very time consuming, but it can also be a very effective way to create a WBS. Project managers often use the bottom-up approach for projects that represent entirely new systems or approaches to doing a job, or to help create buy-in and synergy with a project team.

## Mind Mapping

Some project managers like to use mind mapping to help develop WBSs. As described in Chapter 4 during the discussion of SWOT analysis, mind mapping is a technique that uses branches radiating from a core idea to structure thoughts and ideas. Instead of writing down tasks in a list or immediately trying to create a structure for tasks, mind mapping allows people to write and even draw pictures of ideas in a nonlinear format. This more visual, less structured approach to defining and then grouping tasks can unlock creativity among individuals and increase participation and morale among teams.<sup>8</sup>

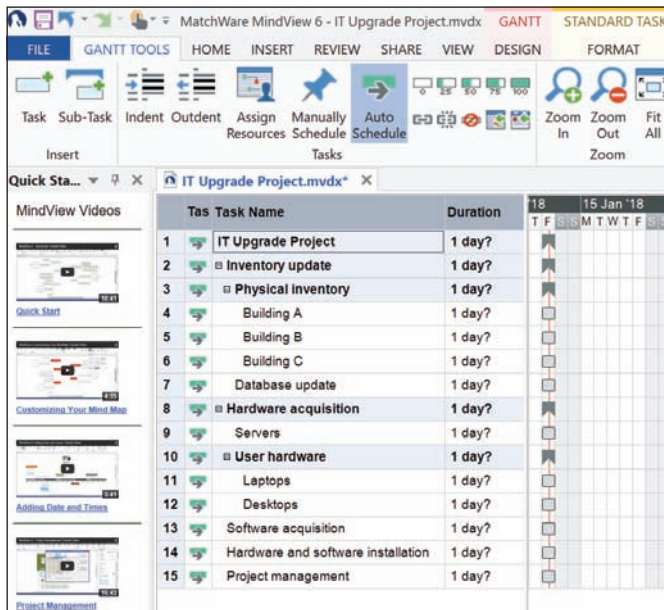
Figure 5-7 shows a diagram created with MatchWare's MindView 6.0 software that uses mind mapping to create a WBS for the IT upgrade project from the opening case. The rectangle near the center represents the entire project. Each of the four main branches radiating from the center represents the main tasks or Level 2 items for the WBS. Different people at the meeting who are creating this mind map might have different roles in the project, which could help in deciding the tasks and WBS structure. For example, Kim would want to focus on all of the project management tasks, and she might also know that they will be tracked in a separate budget category. People who are familiar with acquiring or installing hardware and software might focus on that work, and so on. The team would continue to add branches and items until they exhausted ideas for what work needs to be performed.

After discovering WBS items and their structure using the mind-mapping technique, you could then translate the information into chart or tabular form, as described earlier. A feature of MatchWare's MindView 6.0 software is that you can click a single icon to convert a mind map into a Gantt chart. The mind map provides the task list based on the WBS. MindView also lets you enter information about tasks, such as dependencies and durations, to generate a complete Gantt chart. You can also export your mind map into Microsoft Project. The WBS is



Source: MatchWare MindView 6.0

**FIGURE 5-7** Sample mind-mapping technique for creating a WBS



Source: MatchWare MindView 6.0

**FIGURE 5-8** Gantt chart with WBS generated from a mind map

entered in the Task List column, and the structure is created automatically based on the mind map. Figure 5-8 shows the resulting Gantt chart for the IT upgrade project in MindView 6.0.

Mind mapping can be used for developing WBSs using the top-down or bottom-up approach. For example, you could conduct mind mapping for an entire project by listing the project in the center of a document, adding the main categories on branches radiating from the center, and then adding branches for appropriate subcategories. You could also develop a separate mind-mapping diagram for each deliverable and then merge them to create one large diagram for the entire project. You can also add items anywhere on a mind-mapping document without following a strict top-down or bottom-up approach. After the mind-mapping documents are complete, you can convert them into a chart or tabular WBS form.



## ADVICE FOR YOUNG PROFESSIONALS

As mentioned in this chapter, it is very difficult to create a good WBS. Try to attend meetings in your organization where teams work together to create a WBS. If that's not possible, ask to see WBSs for projects that have been completed or are in process where you work. You can also do your own research to find examples of different WBSs. There are many template files available for Microsoft Project and other project management software. Don't be afraid to ask questions. There is no one way to organize the work that needs to be done on a project. Remember to not leave anything essential out of a WBS, and do not include anything that is superfluous.

## The WBS Dictionary

Many of the items listed on the sample WBSs are rather vague. What exactly does “Database update” mean, for example? The person responsible for this task might think that it does not need to be broken down any further, which could be fine. However, the task should be described in more detail so everyone has the same understanding of what it involves. What if someone else has to perform the task? What would you tell that team member to do? What will it cost to complete the task? More detailed information is needed to answer these and other questions.

A **WBS dictionary** is a document that provides detailed information about each WBS item. The term *dictionary* should not be confused with defining terms or acronyms in this case; such definitions belong in a glossary that would be included elsewhere in the project documentation. Instead, the WBS dictionary is a definition of the work involved in the task—a clarification that makes the summary description in the WBS easier to understand in terms of the approach taken to complete the work.

The format of the WBS dictionary can vary based on project needs. It might be appropriate to have a short paragraph describing each work package. For a more complex project, an entire page or more might be needed for each of the work package descriptions. Some projects might require that each WBS item describe the responsible organization, resource requirements, estimated costs, dependencies with other activities, and other information. Project teams often review WBS dictionary entries from similar tasks to get a better idea of how to create their entries.

In the IT upgrade project described in the opening case, Kim should work with her team and sponsor to determine the level of detail needed in the WBS dictionary. They should also decide where this information will be entered and how it will be updated. Kim and her team decided to enter all of the WBS dictionary information into their enterprise project management system, following departmental guidelines. Table 5-6 is an example of one entry.

The approved project scope statement and its associated WBS and WBS dictionary form the scope baseline. Performance in meeting project scope goals is based on this scope baseline.

**TABLE 5-6** Sample WBS dictionary entry

WBS Dictionary Entry March 20
<b>Project Title:</b> Information Technology (IT) Upgrade Project
<b>WBS Item Number:</b> 2.2
<b>WBS Item Name:</b> Database Update
<b>Description:</b> The IT department maintains an online database of hardware and software on the corporate intranet. We need to make sure that we know exactly what hardware and software employees are currently using and if they have any unique needs before we decide what to order for the upgrade. This task will involve reviewing information from the current database, producing reports that list each department's employees

(continued)



**TABLE 5-6** Sample WBS dictionary entry (*continued*)

and location, and updating the data after performing the physical inventory and receiving inputs from department managers. Our project sponsor will send a notice to all department managers to communicate the importance of this project and this particular task. In addition to general hardware and software upgrades, the project sponsors will ask the department managers to provide information for any unique requirements they might have that could affect the upgrades. This task also includes updating the inventory data for network hardware and software. After updating the inventory database, we will send an e-mail to each department manager to verify the information and make changes online as needed. Department managers will be responsible for ensuring that their people are available and cooperative during the physical inventory. Completing this task is dependent on WBS Item Number 2.1, Physical Inventory, and must precede WBS Item Number 3.0, Hardware and Software Acquisition.

### Advice for Creating a WBS and a WBS Dictionary

As stated previously, creating a good WBS is no easy task and usually requires several iterations. Often, it is best to use a combination of approaches to create a project's WBS. Some basic principles, however, apply to creating any good WBS and its WBS dictionary.

- A unit of work should appear in only one place in the WBS.
- The work content of a WBS item is the sum of the WBS items below it.
- A WBS item is the responsibility of only one person, even though many people might be working on it.
- The WBS must be consistent with the way work actually will be performed; it should serve the project team first, and serve other purposes only if practical.
- Project team members should be involved in developing the WBS to ensure consistency and buy-in.
- Each WBS item must be documented in a WBS dictionary to ensure accurate understanding of the scope of work included and not included in that item.
- The WBS must be a flexible tool to accommodate inevitable changes while properly maintaining control of the work content in the project according to the scope statement.<sup>9</sup>

## VALIDATING SCOPE

It is difficult to create a good project scope statement and WBS for a project. It is even more difficult, especially on IT projects, to verify the project scope and minimize scope changes. Some project teams know from the start that the scope is very unclear and that they must work closely with the project customer to design and produce various deliverables. In this case, the project team must develop a process for scope validation that meets unique project needs. Careful procedures must be developed to ensure that customers are getting what they want and that the project team has enough time and money to produce the desired products and services.



Even when the project scope is fairly well defined, many IT projects suffer from **scope creep**—the tendency for project scope to keep getting bigger and bigger. There are many horror stories about IT projects failing due to problems such as scope creep, including a few classic examples in the following What Went Wrong? feature. For this reason, it is very important to verify the project scope with users throughout the life of the project and develop a process for controlling scope changes.

Scope creep can also be a good thing, if managed well. Later in this chapter, you can see how Northwest Airlines encouraged scope changes on its ResNet project and managed them well.



## WHAT WENT WRONG?

A project scope that is too broad and grandiose can cause severe problems. Scope creep and an overemphasis on technology for technology's sake resulted in the bankruptcy of a large pharmaceutical firm, Texas-based FoxMeyer Drug. In 1994, the CIO was pushing for a \$65 million system to manage the company's critical operations. He did not believe in keeping things simple, however. The company spent nearly \$10 million on state-of-the-art hardware and software and contracted the management of the project to a prestigious (and expensive) consulting firm. The project included building an \$18 million robotic warehouse, which looked like something out of a science fiction movie, according to insiders. The scope of the project kept getting bigger and more impractical. The elaborate warehouse was not ready on time, and the new system generated erroneous orders that cost FoxMeyer Drug more than \$15 million in unrecovered excess shipments. In July 1996, the company took a \$34 million charge for its fourth fiscal quarter, and by August of that year, FoxMeyer Drug filed for bankruptcy.<sup>10</sup>

Another classic example of scope creep comes from McDonald's Restaurants. In 2001, the fast-food chain initiated a project to create an intranet that would connect its headquarters with all of its restaurants and provide detailed operational information in real time. For example, headquarters would know if sales were slowing or if the grill temperature was correct in every single store—all 30,000 of them in more than 120 countries. McDonald's would not divulge detailed information, but they admitted that the project was too large in scale and scope. After spending \$170 million on consultants and initial implementation planning, McDonald's realized that the project was too much to handle and terminated it.<sup>11</sup>

Another major scope problem on IT projects is a lack of user involvement. A prime example occurred in the late 1980s at Northrop Grumman, which specializes in defense electronics, IT, advanced aircraft, shipbuilding, and space technology. An IT project team there became convinced that it should automate the review and approval process of government proposals. The team implemented a powerful workflow system to manage the whole process. Unfortunately, the end users of the system were aerospace engineers who preferred to work in a more casual, ad hoc fashion. They dubbed the system "Naziware" and refused to use it. This example illustrates an IT project that wasted millions of dollars developing a system that was not in touch with the way end users did their work.<sup>12</sup>

(continued)

Failing to follow good project management processes and use off-the-shelf software also results in scope problems. 21st Century Insurance Group in Woodland Hills, California, paid Computer Sciences Corporation \$100 million to develop a system for managing business applications, including managing insurance policies, billing, claims, and customer service. After five years, the system was still in development and supported less than 2 percent of the company's business. Joshua Greenbaum, an analyst at Enterprise Applications Consulting, called the project a "huge disaster" and questioned the insurance company's ability "to manage a process that is pretty well known these days. . . . I'm surprised that there wasn't some way to build what they needed using off-the-shelf components and lower their risk."<sup>13</sup>

**Scope validation** involves formal acceptance of the completed project deliverables. This acceptance is often achieved by a customer inspection and then sign-off on key deliverables. To receive formal acceptance of the project scope, the project team must develop clear documentation of the project's products and procedures to evaluate whether they were completed correctly and satisfactorily. Recall from Chapter 4 that configuration management specialists identify and document the functional and physical characteristics of the project's products, record and report the changes, and audit the products to verify conformance to requirements. To minimize scope changes, it is crucial to do a good job of configuration management and validating project scope.

The project management plan, project documents, verified deliverables, and work performance data are the main inputs for scope validation. The main tools for performing scope validation are inspection and decision-making techniques. The customer, sponsor, or user inspects the work after it is delivered and decides if it meets requirements. The main outputs of scope validation are accepted deliverables, change requests, work performance information, and project documents updates. For example, suppose that Kim's team members deliver upgraded computers to users as part of the IT upgrade project. Several users might complain because the computers did not include special keyboards they need for medical reasons. Appropriate people would review this change request and take appropriate corrective action, such as getting sponsor approval for purchasing the special keyboards.



## GLOBAL ISSUES

Many countries have had difficulties controlling the scope of large projects, especially those that involve advanced technologies and many different users. For example, the state government of Victoria, Australia, introduced a public transportation smart card, called myki, in 2010. Public Transport Victoria's description of the card included the following information: "Many cities around the world have public transport smart cards. myki has been designed to fit our State's unique needs. myki users enjoy an integrated ticketing system that works across the state on trains, trams and buses."<sup>14</sup>

(continued)

Unfortunately, there were many problems in developing and implementing the smart cards. The \$1.35 billion system became valid on all forms of Melbourne public transportation in July 2010, three years and five months after it was meant to replace the Metcard. Users' initial reactions to the myki smart card were mixed, with several reports of myki readers not working on trams. Many skeptics said they would wait until problems were fixed before trying the new system.<sup>15</sup> Many articles described problems with the myki card, revealing obvious difficulty in validating the scope of this high-visibility project. The Public Transport Users Association (PTUA) compiled a long list of problems with the new card and suggested that people stick to using the old card for a while. Clearly, the new system did not meet user requirements and had major flaws. In January 2012, over 18 months after the myki rollout, 70 percent of users still used the old Metcard. The government decided to keep myki in June 2011 after estimating that it would cost taxpayers more than \$1 billion to scrap the troubled system. The Metcard was planned to be phased out for good by the end of 2012. PTUA president Daniel Bowen was not certain that the transition would go smoothly. "To a certain extent it makes sense to bring people across (to myki) because you can't have two systems running at the same time forever. . . . Once you get the majority of people using myki though, they'd better make sure it's humming, otherwise there could be chaos."<sup>16</sup> Myki continued to make headlines in 2015, but not in a good way. "Two years after myki became the only ticket in town, frustrated commuters say the system is still riddled with issues."<sup>17</sup>

## CONTROLLING SCOPE

As you learned in Chapter 4 when you studied integrated change control, change is inevitable on projects, especially changes to the scope of IT projects. Scope control involves managing changes to the project scope while keeping project goals and business strategy in mind. Users often are not sure how they want screens to look or what functionality they will need to improve business performance. Developers are not exactly sure how to interpret user requirements, and they also have to deal with constantly changing technologies.

The goal of scope control is to influence the factors that cause scope changes, to ensure that changes are processed according to procedures developed as part of integrated change control, and to manage changes when they occur. You cannot do a good job of controlling scope if you do not first do a good job of collecting requirements, defining scope, and validating scope. How can you prevent scope creep when you have not agreed on the work to be performed and your sponsor has not validated that the proposed work is acceptable? You also need to develop a process for soliciting and monitoring changes to project scope. Stakeholders should be encouraged to suggest changes that will benefit the overall project and discouraged from suggesting unnecessary changes.

The project management plan, project documents, work performance data, and organizational process assets are the main inputs to scope control. An important tool for performing scope control is data analysis, including variance analysis. **Variance** is

the difference between planned and actual performance. For example, if a supplier was supposed to deliver five special keyboards and you received only four, the variance would be one keyboard. The outputs of scope control include work performance information, change requests, project management plan updates, and project documents updates.

Table 1-2 in Chapter 1 lists the top 10 factors that help IT projects succeed. Four of these ten factors are related to scope validation and control: executive sponsorship, user involvement, optimization, and clear business objectives. To avoid project failures, therefore, it is crucial for IT project managers and their teams to improve user input and executive support and reduce incomplete and changing requirements.

The following sections provide more suggestions for improving scope management on IT projects.

### Suggestions for Improving User Input

Lack of user input leads to problems with managing scope creep and controlling change. How can you manage this important issue? The following suggestions can help a project team improve user input:

- Develop a good project selection process for IT projects. Insist that all projects have a sponsor from the user organization. The sponsor should not work in the IT department, nor should the sponsor be the project manager. Project information, including the project charter, project management plan, project scope statement, WBS, and WBS dictionary, should be easily available in the organization. Making basic project information available will help avoid duplication of effort and ensure that the most important projects are the ones on which people are working.
- Have users on the project team. Some organizations require project managers to come from the business area of the project instead of the IT group. Some organizations assign co-project managers to IT projects, one from IT and one from the main business group. Users should be assigned full time to large IT projects and part-time to smaller projects. A key success factor in Northwest Airlines' ResNet project was training reservation agents—the users—how to write programming code for their new reservation system. (See the Companion website for this text to read the entire case study for the ResNet project.) Because the sales agents had intimate knowledge of the business, they provided excellent input and actually created most of the software.
- Have regular meetings with defined agendas. The idea of meeting regularly sounds obvious, but many IT projects fail because the project team members do not have regular interaction with users. They assume that they understand what users need without getting direct feedback. To encourage this interaction, users should sign off on key deliverables presented at meetings.
- Deliver something to project users and sponsors on a regular basis. If the delivered product is hardware or software, make sure it works first.
- Do not promise to deliver what the team cannot deliver in a particular time frame. Make sure the project schedule allows enough time to produce the deliverables.

- Locate users with the developers. People often get to know each other better by being in close proximity. If the users cannot be physically moved to be near developers during the entire project, they should set aside certain days to work in the same location.

## Suggestions for Reducing Incomplete and Changing Requirements

Some requirement changes are expected on IT projects, but many projects have too many changes to their requirements, especially during later stages of the project life cycle when it is more difficult to implement changes. The following suggestions can help improve the requirements process:

- Develop and follow a requirements management process that includes procedures for determining initial requirements.
- Employ techniques such as prototyping, use case modeling, and Joint Application Design to understand user requirements thoroughly. **Prototyping** involves developing a working replica of the system or some aspect of the system. These working replicas may be throwaways or an incremental component of the deliverable system. Prototyping is an effective tool for gaining an understanding of requirements, determining the feasibility of requirements, and resolving user interface uncertainties. **Use case modeling** is a process for identifying and modeling business events, who initiated them, and how the system should respond to them. It is an effective tool for understanding requirements of information systems. **Joint Application Design (JAD)** uses highly organized and intensive workshops to bring together project stakeholders—the sponsor, users, business analysts, programmers, and so on—to jointly define and design information systems. These techniques also help users become more active in defining system requirements. Consult a systems analysis and design text for details on these techniques.
- Put all requirements in writing and keep them current and readily available. Several tools are available to automate this function. For example, a type of software called a requirements management tool aids in capturing and maintaining requirements information, provides immediate access to the information, and assists in establishing necessary relationships between requirements and information created by other tools.
- Create a requirements management database for documenting and controlling requirements. Computer Aided Software Engineering (CASE) tools or other technologies can assist in maintaining a repository for project data. A CASE tool's database can also be used to document and control requirements.
- Provide adequate testing to verify that the project's products perform as expected. Conduct testing throughout the project life cycle. Chapter 8, Project Quality Management, includes more information on testing.
- Use a process for reviewing requested requirements changes from a systems perspective. For example, ensure that project scope changes include associated cost and schedule changes. Require approval via signatures of appropriate stakeholders. It is crucial for the project manager to lead the team

in its focus on achieving approved scope goals and not getting sidetracked into doing additional work. For example, in his book *Alpha Project Managers*, Andy Crowe tried to uncover what the best or “alpha” project managers do differently from other project managers. One of these alpha project managers explained how he learned an important lesson about scope control:

Toward the end of some projects I’ve worked on, the managers made their teams work these really long hours. After the second or third time this happened, I just assumed that this was the way things worked. Then I got to work with a manager who planned everything out really well and ran the team at a good pace the whole time, and we kept on schedule. When the customer found out that things were on schedule, he kept trying to increase the scope, but we had a good manager this time, and she wouldn’t let him do it without adjusting the baselines. That was the first time I was on a project that finished everything on time and on budget, and I was amazed at how easy she made it look.<sup>18</sup>

- Emphasize completion dates. For example, a project manager at Farmland Industries, Inc. in Kansas City, Missouri, kept her 15-month, \$7 million integrated supply-chain project on track by setting the project deadline. She said, “May 1 was the drop-dead date, and everything else was backed into it. Users would come to us and say they wanted something, and we’d ask them what they wanted to give up to get it. Sticking to the date is how we managed scope creep.”<sup>19</sup>
- Allocate resources specifically for handling change requests. For example, Peeter Kivestu and his ResNet team at Northwest Airlines knew that users would request enhancements to the reservations system they were developing. They provided a special function key on the ResNet screen for users to submit their requests, and the project included three full-time programmers to handle these requests. Users made over 11,000 enhancement requests. The managers who sponsored the four main software applications had to prioritize the software enhancement requests and decide as a group what changes to approve. The three programmers then implemented as many items as they could, in priority order, given the time they had. Although they only implemented 38 percent of the requested enhancements, they were the most important ones, and the users were very satisfied with the system and process.

## USING SOFTWARE TO ASSIST IN PROJECT SCOPE MANAGEMENT

---

Project managers and their teams can use several types of software to assist in project scope management. As shown in several of the figures and tables in this chapter, you can use word-processing software to create scope-related documents, and most people use spreadsheet or presentation software to develop various charts, graphs, and matrixes related to scope management. Mind-mapping software can be useful in developing a WBS.

Project stakeholders also transmit project scope management information using various types of communication software such as e-mail and assorted web-based applications.

Project management software helps you develop a WBS, which serves as a basis for creating Gantt charts, assigning resources, allocating costs, and performing other tasks. You can also use the templates that come with various project management software products to help you create a WBS for your project.

You can also use many types of specialized software to assist in project scope management. Many IT projects use special software for requirements management, prototyping, modeling, and other scope-related work. Because scope is such a crucial part of project management, many software products are available to assist in managing project scope. For example, Gartner estimates that the market for requirements definition and management tools was \$280 million in 2014 and growing over 3 percent annually. New tools focus on improving collaboration and speed to better address mass market needs.<sup>20</sup>

## CONSIDERATIONS FOR AGILE/ADAPTIVE ENVIRONMENTS

---

The *PMBOK® Guide – Sixth Edition* provides the following information for project scope management:

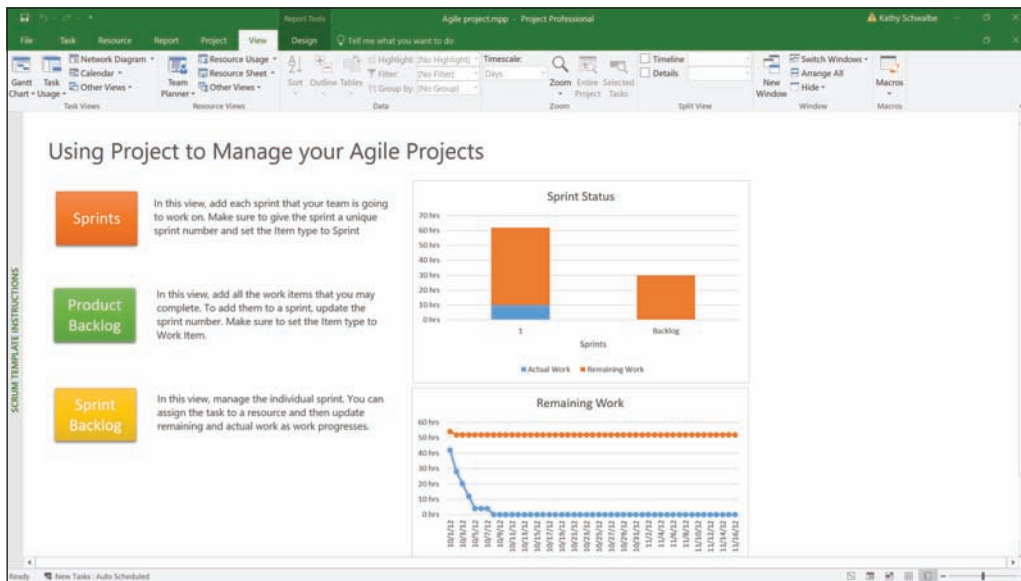
In projects with evolving requirements, high risk, or significant uncertainty, the scope is often not understood at the beginning of the project or it evolves during the project. Agile methods deliberately spend less time trying to define and agree on scope in the early stage of the project and spend more time establishing the process for its ongoing discovery and refinement. Many environments with emerging requirements find that there is often a gap between the real business requirements and the business requirements that were originally stated. Therefore, agile methods purposefully build and review prototypes and release versions in order to refine the requirements. As a result, scope is defined and redefined throughout the project. In agile approaches, the requirements constitute the backlog.<sup>21</sup>

As discussed in Chapter 2, stakeholders define and approve the detailed scope before the start of an iteration with an adaptive or agile product life cycle, producing a usable product at the end of each iteration. Does this mean that scope does not change for other product life cycles? As described in this chapter, the high-level requirements should always be defined early (in the project charter and initial scope statement), but the detailed scope develops over time. For example, in a large, traditional project like building a house, the owners set the high-level requirements (number of square feet, location, basic blueprint, etc.), but detailed requirements are defined later (which particular flooring, what model number for appliances, etc.). Also, remember a key difference between an agile approach and others is that it provides *several* usable products during the project, while other approaches only deliver the final product at the end of the project.

You can still use project management software for agile projects. For example, Figure 5-9 shows a screenshot of a Microsoft Project 2016 template called “Agile Project Management.” It includes links and brief instructions for creating sprints, the product backlog, and the sprint backlog. You can also create other charts, as shown in the figure.

Project scope management is very important, especially on IT projects. After selecting projects, organizations must plan scope management, collect the requirements and define





**FIGURE 5-9** Using Microsoft Project 2016 for agile projects

the scope of the work, break down the work into manageable pieces, validate the scope with project stakeholders, and manage changes to project scope. Using the basic project management concepts, tools, and techniques discussed in this chapter can help you manage project scope successfully.

## CASE WRAP-UP

Kim Nguyen reviewed guidelines for creating WBSs that were provided by her company and other sources. She had a meeting with the three team leaders for her project to get their input on how to proceed. They reviewed several sample documents and decided to have major groupings for their project based on updating the inventory database, acquiring the necessary hardware and software, installing the hardware and software, and performing project management. After they decided on a basic approach, Kim led a meeting with the entire project team of 12 people, with some attending virtually. She reviewed the project charter and stakeholder register, described the basic approach they would use to collect requirements and define the project scope, and reviewed sample WBSs. Kim opened the floor for questions, which she answered confidently. She then let each team leader work with his or her people to start writing the detailed scope statement and their sections of the WBS and WBS dictionary. Everyone participated in the meeting, sharing their expertise and openly asking questions. Kim could see that the project was off to a good start.



## Chapter Summary

---

Project scope management includes the processes to ensure that the project addresses all the work required to complete the project successfully. The main processes include planning scope management, collecting requirements, defining scope, creating the WBS, validating scope, and controlling scope.

The first step in project scope management is planning scope management. The project team reviews information and uses expert judgment and meetings to help create a scope management plan and requirements management plan.

The next step is collecting requirements, a crucial part of many IT projects. It is important to review the project charter and meet with key stakeholders listed in the stakeholder register when collecting requirements. The main outputs of this process are requirements documentation and a requirements traceability matrix.

A project scope statement is created in the scope definition process. This document often includes a product scope description, product user acceptance criteria, detailed information on all project deliverables, and information on project boundaries, constraints, and assumptions. There are often several versions of the project scope statement to keep scope information detailed and up to date.

A work breakdown structure (WBS) is a deliverable-oriented grouping of the work involved in a project that defines its total scope. The WBS forms the basis for planning and managing project schedules, costs, resources, and changes. You cannot use project management software without first creating a good WBS. A WBS dictionary is a document that provides detailed information about each WBS item. A good WBS is often difficult to create because of the complexity of the project. There are several approaches for developing a WBS, including using guidelines, the analogy approach, the top-down approach, the bottom-up approach, and mind mapping.

Validating scope involves formal acceptance of the completed project deliverables.

Controlling scope involves controlling changes to the project scope.

Poor project scope management is one of the key reasons projects fail. For IT projects, it is important for good project scope management to have strong user involvement, executive support, a clear statement of requirements, and a process for managing scope changes.

Many software products are available to assist in project scope management. The WBS is a key concept in properly using project management software because it provides the basis for entering tasks.

Be sure to consider how project scope management can differ in agile/adaptive environments.

## Discussion Questions

---

1. What is involved in project scope management, and why is good project scope management so important on IT projects?
2. What is involved in collecting requirements for a project? Why is it often difficult to do?
3. Discuss the process of defining project scope in more detail as a project progresses, going from information in a project charter to a project scope statement, WBS, and WBS dictionary.

4. Describe different ways to develop a WBS and explain why it is often so difficult to do.
5. What is the main technique used for validating scope? Give an example of scope validation on a project.
6. Using examples in this book or online, describe a project that suffered from scope creep. Could it have been avoided? How? Can scope creep be a good thing? When? What can organizations do to successfully manage inevitable changes in scope that are good for business?
7. Why do you need a good WBS to use project management software? What other types of software can you use to assist in project scope management?
8. How is scope management different on agile projects?

## Quick Quiz

---

1. \_\_\_\_\_ refer(s) to all the work involved in creating the products of the project and the processes used to create them.
  - a. Deliverables
  - b. Milestones
  - c. Scope
  - d. Product development
2. Which tool or technique for collecting requirements is often the most expensive and time consuming?
  - a. Interviews
  - b. Focus groups
  - c. Surveys
  - d. Observation
3. A \_\_\_\_\_ is a deliverable-oriented grouping of the work involved in a project that defines its total scope.
  - a. scope statement
  - b. WBS
  - c. WBS dictionary
  - d. work package
4. What approach to developing a WBS involves writing down or drawing ideas in a nonlinear format?
  - a. Top-down
  - b. Bottom-up
  - c. Analogy
  - d. Mind mapping

5. Assume that you have a project with major categories called planning, analysis, design, and testing. What level of the WBS would these items fall under?
  - a. 0
  - b. 1
  - c. 2
  - d. 3
6. Which of the following is not a best practice that can help in avoiding scope problems on IT projects?
  - a. Keep the scope realistic.
  - b. Use off-the-shelf hardware and software whenever possible.
  - c. Follow good project management processes.
  - d. Don't involve too many users in scope management.
7. What constitutes requirements for agile projects?
  - a. The Scrum
  - b. The backlog
  - c. The sprints
  - d. The requirements list
8. Scope \_\_\_\_\_ is often achieved by a customer inspection and then sign-off on key deliverables.
  - a. acceptance
  - b. validation
  - c. completion
  - d. close-out
9. Which of the following is not a suggestion for improving user input?
  - a. Develop a good project selection process for IT projects.
  - b. Have users on the project team.
  - c. Co-locate users with developers.
  - d. Only have meetings as needed, not on a regular basis.
10. Project management software helps you develop a \_\_\_\_\_, which serves as a basis for creating Gantt charts, assigning resources, and allocating costs.
  - a. project plan
  - b. schedule
  - c. WBS
  - d. deliverable

## Quick Quiz Answers

---

1. c; 2. a; 3. b; 4. d; 5. c; 6. d; 7. b; 8. b; 9. d; 10. c

## Exercises

---

1. You are working on a project to develop a new or enhanced system to help people at your college, university, or organization to find jobs. The system must be tailored to your student or work population and be very easy to use. Write a short paper describing how you would collect requirements for this system, and include at least five requirements in a requirements traceability matrix.
2. Read the report cited in the chapter called PMI's *Pulse of the Profession: Requirements Management—A Core Competency for Project and Program Success*. Find one or two other reports on requirements management written in the past year. Summarize the findings of the reports and your opinion on the topic.
3. Use MindView or similar software to develop a WBS for a project of your choice. You can use one of the examples in this chapter, or create your own. Take screenshots and print out the WBS in various formats (mind map, chart, and tabular).
4. Create the same WBS from Exercise 3 using Microsoft Project or similar software and indenting categories appropriately. Use the outline numbering feature to display the outline numbers, or enter the numbers manually. Do not enter any durations or dependencies. See Appendix A or Project 2016's Help for instructions on creating a WBS. Print the resulting Gantt chart on one page, and be sure to display the entire Task Name column.
5. Using the file you created in MindView or similar software for Exercise 3, export the WBS into Project 2016 or similar software. Make adjustments as needed and then print the Gantt chart from Project 2016 or similar software. Also write a paragraph describing your opinion of using a mind map versus just entering information in the Task Name column.
6. Create a WBS for one of the following projects:
  - Introducing self-checkout registers at your school's bookstore
  - Providing a new Internet cafe onsite at your organization
  - Developing an app that you and a friend will create and sell online within three months
  - Earning your CAPM® or PMP® certification

Decide on all of the Level 2 categories for the WBS. Use your choice of software to create a WBS in chart form and as tasks in a Gantt chart. Do not enter any durations or dependencies. Be sure to include all of the deliverables and work required for the project. Make notes of questions you had while completing this exercise.

7. Review three different template files from Microsoft Project 2016, MindView, or any other project management software tool. What do you think about the WBSs? Write a short paper summarizing your analysis, and provide at least three suggestions for improving one of the WBSs.
8. Research the benefits of and requirements for earning a certification in business analysis, as described in the What Went Right feature in this chapter. Summarize your findings in a short paper.

## Running Case

---

You have been selected as the project manager for the Global Treps project. You helped to run a local shark tank like event at your college last year as part of a class project, so you have a general idea of what is involved. The schedule goal is six months, and the budget is \$120,000. Your favorite professor, Dr. K., and a few of her associates have agreed to fund the project. Your strengths are your organizational and leadership skills.

You are a senior, live on-campus, and get free room and board by being a resident assistant in your dorm. Bobby, a computer whiz who funded a lot of his college expenses by building websites, will be your main technical guy on the project. He goes to your college and lives off-campus. Three other people will form your core project team: Kim, a new college grad now working for a non profit group in Vietnam; Ashok, a business student in India; and Alfreda, a student in the United States planning to visit her home town in Ethiopia for two months in a few months. You will hold most meetings virtually, but you can meet face-to-face with Bobby and Dr. K. as needed. You have all known each other for at least a year and are excited to make this project a success.

You and your team members will do the work part-time while you finish school or work at other jobs, but you can use up to \$50,000 total to pay yourselves. You estimate that you will need another \$30,000 for travel expenses, \$20,000 for hardware and software, and the other \$20,000 will go toward organizing events, consultants, legal/business fees, etc.

Your goal is to develop a fully functioning website and test it by holding four events in four different countries. You'll make improvements to the site after those events, develop plans to scale it up, and recommend how to transition the project results into a successful business.

Recall from Chapter 4 that this system would include the following capabilities:

- Provide guidelines and templates for running a shark tank type event
- Accept donations from potential investors targeted toward specific schools or organizations wishing to host an event (similar to the popular DonorsChoose.org site, where people can fund teachers' requests)
- Accept ideas for needed new products or services
- Provide the ability for organizations to create their own custom site to solicit local participants and sharks, accept applications, and promote the winners as well as losers.

Note that you decided *not* to include the idea of providing an online version of the event as part of the initial project as your sponsor and team decided that physical events would be most effective. You have also decided to limit the scope of this first project to provide the ability for 20 organizations to create their own custom websites. Your team members will screen the organizations and assist people in using the site to plan their events.

You plan to hold four shark tank like events within four months, using your team members abroad to help organize and run those events, plus one at your college. Your semester has just started, so you plan to hold your event at the end of the term. The project will fund refreshments

for the events and prizes for the winners, with a budget of \$1,000 for each event. You don't think you'll get any donations via the new website before these events, but you'll try to have it set up to accept donations by the last month.

You will create some short videos to show people how to use the site and provide suggestions for holding the events.

After testing the site and getting customer feedback, you will make some changes and document recommendations for a follow-on project. You will also create a business plan recommending how to transition this project into a real business that can make a profit after two years.

Assume that you would pay for a new website and account through an online provider. Bobby would do most of the customization/programming for the site, but you would consider outsourcing or purchasing services to provide some of the capabilities like accepting donations and developing the short videos on the site. You would also buy a new laptop and Internet access for your three team members abroad so that they could share information with their contacts in those countries.

You and Dr. K. want to attend all of the events as part of the project, and you might include a full face-to-face meeting with the whole team if possible.

## Tasks

---

1. Document your approach for collecting requirements for the project. Include at least five technical requirements and five non technical requirements in a requirements traceability matrix. Also develop a list of at least five questions you would like to ask the project sponsor. Let your instructor answer them for you, if applicable.
2. Develop a first version of a project scope statement for the project. Use the template provided on the Companion website for this text and the example in Chapter 3 as guides. Be as specific as possible in describing product characteristics and requirements, as well as all of the project's deliverables. Be sure to include the Global Treps website, four shark tank like events, a business plan, and outsourced items (laptops, donation acceptance, video creation, etc.) as part of the project scope.
3. Develop a work breakdown structure for the project. Break down the work to Level 3 or Level 4, as appropriate. Use the template on the Companion website and samples in this text as guides. Print the WBS in list form. Be sure the WBS is based on the project charter (created for the Chapter 4 Running Case), the project scope statement created in Task 2 above, and other relevant information.
4. Use the WBS you developed in Task 3 to begin creating a Gantt chart using your choice of software. Do not enter any durations or dependencies. Print the resulting Gantt chart on one page, and be sure to display the entire Task Name column.
5. Develop a strategy for scope validation and change control for this project. Write a short paper summarizing key points of the strategy.

## Key Terms

---

analogy approach	requirements traceability matrix (RTM)
benchmarking	scope
bottom-up approach	scope baseline
decomposition	scope creep
deliverable	scope validation
Joint Application Design (JAD)	top-down approach
project scope management	use case modeling
project scope statement	variance
prototyping	WBS dictionary
requirement	work breakdown structure (WBS)
requirements management plan	work package

## Endnotes

---

- <sup>1</sup> Karl Wiegers, *Software Requirements*, Second edition (Redmond, WA: Microsoft Press, 2003), p. 7.
- <sup>2</sup> Project Management Institute, Inc., *A Guide to the Project Management Body of Knowledge (PMBOK® Guide) – Sixth Edition* (2017), p. 140.
- <sup>3</sup> Amy Hillman, “The Rise in Business-Analytics Degrees,” *Huffington Post* (May 13, 2014).
- <sup>4</sup> Project Management Institute, *Pulse of the Profession®: Requirements Management* (August 2014).
- <sup>5</sup> Craig Smith, “Interview and Book Review: How Google Tests Software,” [www.infoq.com](http://www.infoq.com) (September 11, 2012).
- <sup>6</sup> John Simpson, “2011: The State of Requirements Management” (2011).
- <sup>7</sup> Project Management Institute, *Pulse of the Profession: Requirements Management—A Core Competency for Project and Program Success* (August 2014).
- <sup>8</sup> Mindjet Visual Thinking, “About Mind Maps,” [www.mindjet.com](http://www.mindjet.com) (2002).
- <sup>9</sup> David I. Cleland, *Project Management: Strategic Design and Implementation*, Second edition (New York: McGraw-Hill, 1994).
- <sup>10</sup> Geoffrey James, “Information Technology Fiascoes ... and How to Avoid Them,” *Datamation* (November 1997).
- <sup>11</sup> Paul McDougall, “8 Expensive IT Blunders,” *InformationWeek* (October 16, 2006).
- <sup>12</sup> Op. cit.
- <sup>13</sup> Marc L. Songini, “21st Century Insurance Apps in Limbo Despite \$100M Investment,” *Computerworld* (December 6, 2002).

- <sup>14</sup> [www.ptv.vic.gov.au/tickets/myki/](http://www.ptv.vic.gov.au/tickets/myki/) (accessed March 1, 2012).
- <sup>15</sup> Megan Levy, "Peak-Hour Test for myki Smartcard System," *The Age* (July 26, 2010).
- <sup>16</sup> Greg Thom, "When It Comes to myki, Like It or Lump It," *Herald Sun* (January 20, 2012).
- <sup>17</sup> Alex White, "The \$1.5 Billion myki Debacle Labor Doesn't Want to Know About," *Herald Sun* (January 3, 2015).
- <sup>18</sup> Andy Crowe, *Alpha Project Managers: What the Top 2 percent Know That Everyone Else Does Not* (Kennesaw, GA: Velociteach Press, 2006), pp. 46–47.
- <sup>19</sup> Julia King, "IS Reins in Runaway Projects," *Computerworld* (September 24, 1997).
- <sup>20</sup> Thomas E. Murphy, "Market Guide for Software Requirements Definition and Management Solutions," Gartner (October 7, 2014).
- <sup>21</sup> Project Management Institute, Inc., *A Guide to the Project Management Body of Knowledge (PMBOK® Guide) – Sixth Edition* (2017), p. 133.