

---

## Section IV

### *Domain II: Process*

The *Examination Content Outline* (ECO) specifies that Domain II covers 50% of the exam. The Process domain includes the technical project management skills, methods, and the activities needed to manage a project and deliver the benefits for which the project was undertaken. In this section, you'll find the following chapters:

- Scope
- Schedule
- Budget and Resources
- Quality of Deliverables and Products
- Communications
- Risks and Issues
- Procurement
- Stakeholders

Managing project governance, artifacts, issues, changes, the use and transfer of lessons learned, and product turnover to operations are also part of this domain.

# 7 Scope

## Introduction

You already know that a project must, from start to finish, help to achieve the goals and objectives for which it was selected. Eliciting and analyzing requirements, defining project and product scope, and then building and delivering that scope in accordance with those requirements are all at the heart of this value delivery system.

The goals for delivering project scope are the same regardless of the project life cycle and delivery approach. When managing scope, a project manager must define what work is required and then ensure all that work—and only that work—is completed. This is generally an easy topic, but we all have gaps in our knowledge. Be sure to review the Quicktest and make note of your gaps so you can pay particular attention to those sections in this chapter.

In addition to reviewing the Quicktest, see if the following list helps you uncover gaps in your knowledge.

### TRICKS OF THE TRADE™

#### Things to Know about Scope Management for the Exam

- The project manager must plan how they will determine the scope, as well as how they will manage and control scope. This is part of the scope management plan.
- Scope must be clearly defined and formally approved before work starts. If using an adaptive approach, this may be done at a higher level with a summarized agreement.
- Requirements are elicited from all stakeholders, not just the person who assigned the project.
- Requirements elicitation can take a substantial amount of time, especially on large projects.
- Requirements must be evaluated against the business case, ranked, and prioritized to determine what is in and out of scope.
- A work breakdown structure (WBS) is utilized on all projects that use a predictive approach. Using this tool enables the project manager to clarify identified scope as well as find additional scope.
- A backlog, an agile alternative to a traditional WBS, may be utilized on projects using adaptive approaches. A backlog creates visibility into the scope as well as the overall priorities of the project because a backlog is ranked in priority order.
- While the project is being completed, the project manager must check to make sure all the work included in the project management plan is being done—and only that work.
- Gold plating a project (adding extras) is not allowed.
- Any change to scope must be evaluated for its effect on schedule, cost, risk, quality, resources, and customer satisfaction.
- On plan-driven projects, changes to scope require approval; scope changes should not be approved if they relate to work that does not fit within the project charter.
- Scope priorities and changes are more flexible on agile projects where work is planned and completed iteratively and incrementally, but change is not free. This means that when scope is added, the backlog is reprioritized and earlier prioritized items move below the new scope that has been added. It is possible that some scope on the bottom of the backlog will be pushed to another release or another project to preserve cost and schedule baselines.

## QUICKTEST

- Product Scope
- Project Scope
- Timeboxing
- Minimal viable product (MVP)
- Scope Management process
- Scope management plan
- Requirements management plan
- Product roadmap
- Product backlog
- Requirements elicitation methods
  - Brainstorming
  - Interviewing
  - Focus groups
  - Questionnaires and surveys
  - Voting
  - Multicriteria decision analysis
  - Nominal group technique
  - Observation
  - Prototyping
  - Facilitation
  - Mind maps
  - Context diagrams
  - Affinity diagrams
- Balancing requirements
- Iteration reviews
- Acceptance criteria
- Definition of done
- Requirements traceability matrix
- Product analysis
- Project scope statement
- Work breakdown structure (WBS)
- WBS dictionary
- Scope baseline
- Agile scope decomposition
- Inspection
- Customer-valued prioritization
- Incremental product delivery

- The project manager and the project team should continuously determine what is and is not included in the project scope.
- Internal verification followed by customer acceptance of deliverables happens throughout the project.

## Definitions Related to Scope Management

It's important to understand the distinction between product scope and project scope. Those definitions follow, along with a few more definitions we want to start you off with for understanding the chapter content.

### Product Scope

Product scope can be defined as the product deliverables with their associated features and functions. Another way to say this is: The requirements that relate to the product, service, or result of the project is the product scope. It answers the question, "What end result is needed?" There may be a separate, preliminary project to determine product scope, or the requirements may be defined as part of the project, depending on the needs of the project and the organization.

**Example** Let's say the project is to build a new train terminal. The product scope is "a train terminal that meets these technical specifications." The technical specifications would be complex and comprehensive as defined by qualified subject matter experts. To determine if the project successfully achieved the product scope, the new train terminal is compared to the specifications, which were recorded in the requirements documentation and the project scope statement for the project. All aspects of the train station would need to be tested to ensure they work according to plan before the train station is accepted as complete and turned over to operations.

### Project Scope

The project scope is the work the project team will do to deliver the product scope. It includes the product scope. For the train terminal example, the project scope would be "a train terminal that meets these technical specifications," plus the management and delivery of all the work to deliver the train terminal. In other words, project scope includes the planning, coordination, and management activities that ensure the product scope is achieved. These efforts become part of the scope baseline and scope management plan, which are parts of the project management plan.

### Iteration

On agile projects, this is a specifically set period of time during which a project team refines plans or builds the product of the project. In the context of planning, plans are iterated and refined as new information becomes available. In the context of scope, the team builds the product in increments during fixed periods of time called iterations (or sprints, in Scrum). Iterations are set in specific "timeboxes." (See timeboxing next.)



### Timeboxing

For an agile project a timebox is a short, fixed period of time set for the team to complete a selected and prioritized set of activities. In the context of scope this can translate as the completion of a specific set of stories during a two-week iteration (or sprint), for example. If the work planned for the iteration isn't complete within the two-week iteration, the team leaves the uncompleted work on the backlog to be undertaken during another iteration. So it is the timebox (the iteration) that is honored. There is no "complete all the stories no matter how long it takes" approach.

### Minimal Viable Product (Minimal Marketable Feature or MMF)

The term "MVP," or minimal viable product, refers in agile to an increment of product that is at least useful enough that the customer can potentially take delivery of the MVP and use it while the team continues to build the rest of the product. We say "potentially" here because while the team can show the customer the MVP, the customer can sign off on it and accept delivery of it and use it. Alternatively, the customer can accept it, sign off on it, and wait for additional MVPs to be added before taking delivery. These valuable but partially completed products are also known as minimal marketable features, or MMFs. A product release will typically have several MMFs integrated together, but what is included is up to the customer.

**Example** The customer could take delivery of a skateboard that has the potential to be electric. They could then use it without the electric components, and have it upgraded once the electrical features are ready. They could later decide if they want the already available handles, which are optional, but while they are waiting for the team to build the electrical components for installation, they have the value of using the skateboard in its basic form, its MVP.

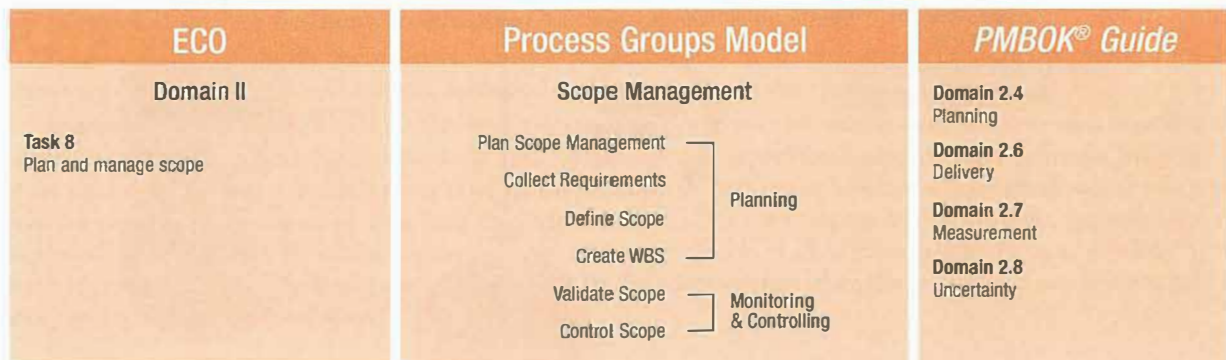


## Scope Management Overview

For the exam, you need to understand the scope management process from the perspective of the Process Groups model as well as from the perspective of an adaptive environment. You also need a solid understanding of how these models fit into the concepts found in the *Examination Content Outline (ECO)* and the *PMBOK® Guide*.

### The *Examination Content Outline (ECO)* and Process Groups Model

The ECO's Process domain shows a task called Plan and Manage Scope, which is analogous to the Process Groups model's Scope Management process. This is illustrated in the first two columns of the following chart.



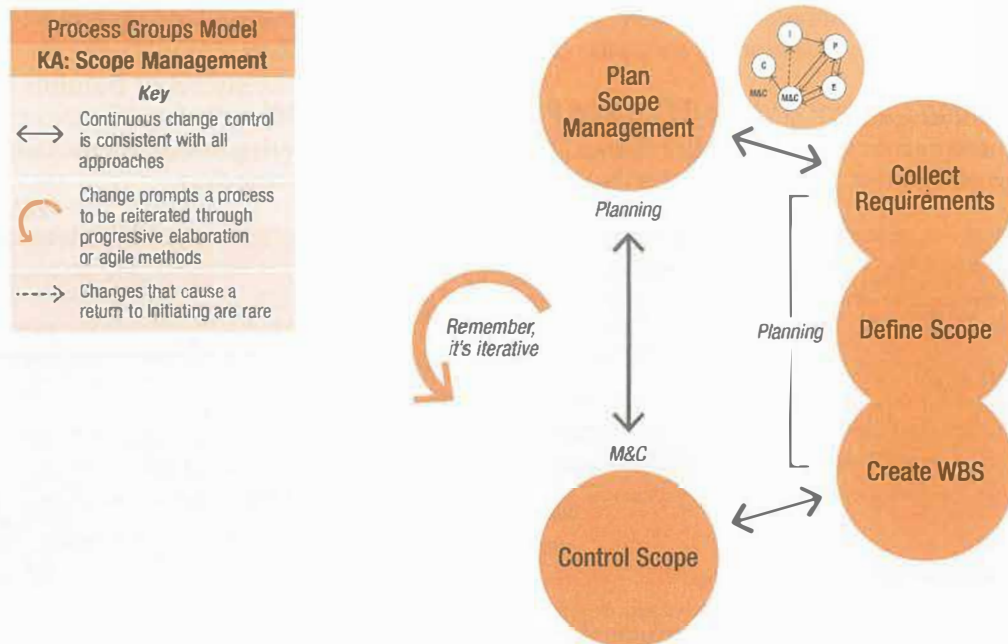
Now, only one task is listed in the ECO column above. Does this mean that only “Plan and manage scope” in the ECO is relevant to the Process Groups model's planning and monitoring and controlling of scope, regardless of the project attributes and the selected project life cycle? Of course not. It would help you to hold the ECO in your hands right now or to have it open on your computer as you work through this section of the chapter. Thinking holistically, you may have already identified that the ECO task, “plan and manage quality of products/deliverables” is intimately tied to managing scope. The team also uses the schedule and budget to create the scope.

First, remember that many or all People domain skills are used to manage all project constraints, since the project manager works with others to get things done. Now, review the following examples and skim through the ECO to think about how these and other tasks fit together with the plan and manage scope task. Think about the ECO tasks holistically. These are just a few examples of other ECO tasks helpful in managing scope:

- Scope management relies on the project manager's work to ensure that team members/stakeholders are adequately trained (domain I, task 5). Regardless of whether there are generalizing specialists for an agile project or specialist team members for a plan-driven project, the project manager identifies what training is needed and provides it as part of supporting team performance and addressing and removing impediments (People domain, tasks 4 and 7).
- Look at the ECO's Business Environment domain, task 2: Evaluate and deliver project benefits and value. This is about balancing the constraints of cost, time, and quality in order to build product scope.

Requirements for the most part are defined early in a plan-driven project, while on an agile project it is understood that scope emerges and more requirements gathering with stakeholders is necessary over time. Figure 7.1 is a visualization of scope management at a high level from the Process Groups model perspective. It can help you visualize where you are

in the scope management process as you continue with this chapter, and understanding it will help you on the exam as you read scenario questions.



**FIGURE 7.1** Scope management process

### Plan-driven Scope Management

There are a lot of acceptable ways to manage scope. If you do it differently than described here, you are not necessarily wrong. For the exam think of the scope management process as including the following for predictive project management:

1. A plan is developed for how to plan, validate, and control scope and requirements.
2. Determine requirements. Ensure they support the project's business case (as described in the project charter)—the benefits and value the project is meant to deliver (Business Environment domain, task 2).
3. Analyze and balance stakeholder needs to determine scope.
4. Create a WBS to break the scope down to smaller, more manageable pieces. Define each piece in a WBS dictionary.
5. Obtain validation (signed acceptance) that the completed scope of work is acceptable to the customer or sponsor.
6. Measure scope performance and adjust as needed.

No one can request or add work that is not related to the reason for initiating the project. Yet, in your real world, do people want work done and try to attach it to any project they can to get that work accomplished? Do you see scope on projects that doesn't support the company's business objectives? It happens all the time. Change control is about protecting the project from unapproved changes in a predictive environment.

When taking the exam, assume you have the authority in a predictive environment to say no when someone tries to add unrelated scope to your project. This is important to internalize because so many of us do not have this authority in the real world.

Note that in a predictive environment, creating a work breakdown structure (WBS) is a required part of project management. If you have never created one or do not currently use a WBS on your projects, this chapter will help you understand how beneficial this tool is and what it can do for you. Remember, the exam asks questions at an expert level and assumes you have experience using various tools.

### Agile Scope Management

In adaptive environments, scope management looks like this:

1. Requirements are identified and documented at a sufficient level of detail so they can be prioritized and estimated at a high level.
2. The product's features—collectively the product scope—are kept in a list called the product backlog.
3. The work is broken into product releases. A project may include one or more releases.
4. For each release, the work is completed through iterations (specifically defined periods of time, like two weeks, or three weeks, for example).
5. The work of each iteration (and release) is defined successively in more detail just before the work for each iteration begins. In this way, decisions are deferred to “the last responsible moment.”

Product scope is typically more flexible in agile projects than it is for plan-driven projects. As essential product features are delivered in early agile releases, more optional features may be deferred.

**Example** You are buying a new, custom-made electric car. You travel a lot, so you want the front seats to recline into beds so you can pull into a campground to sleep. Your budget doesn't support these adaptive seats right now, but you need the car right away. Neither your budget nor timeline support the self-driving features that will eventually allow you to nap while the car drives autonomously on a long road trip. So the electric car company builds the basic car and delivers it quickly in Release 1. Later, for another release, you return for installation of a few more custom features you have agreed upon with the car builder. Eventually you will come back for the adaptive seats when the price goes down and you have the money and time for their installation.

#### TRICKS OF THE TRADE

Early agile practitioners came from the software development industry. They were keenly aware that most software users actually use only a small percentage of the features of a given software package. So they thought *why wait for the entire package to be built when the most valuable features can be used right away* (while delivering a large feature set is notoriously difficult to do on time)? They recognized what became an agile tenet: a distinct advantage of an adaptive environment is the value saved on the features not built. In other words, part of delivering value is deciding on what work is not done (immediately, or possibly ever).

In our example above, what if later you decided your car did not need seats that reclined into beds? You could cancel that feature and just stay with the basic car. If you did decide to get that feature later though, the chances may be good that you will be buying an improved feature that has also become more economical.

**Note:** For a plan-driven project life cycle you will definitely need a WBS, while in a more change-driven life cycle you may have a WBS, the product backlog may take its place, or you may have both. Look for evidence in exam questions that tells you which type of life cycle you are dealing with.

### Desired Outcomes of Scope Management

Assume for the exam that scope is properly planned and managed unless information in an exam question indicates otherwise. This means that the following outcomes should be expected as a result of scope management:

- Throughout the project, the project team has a clear understanding of product and project requirements or they have the ability to make those requirements clear through interactions with the project manager and stakeholders.
- Throughout the project, data are gathered and earned value measurement is performed to indicate project progress relative to plan. This allows the team and stakeholders to best manage target results and make adjustments where necessary. This may take the form of traditional earned value measurement for plan-driven projects (see the “Budget and Resources” chapter). For agile projects, teams may use burnup and burndown charts (see the “Build and Support Performance” chapter).
- Project scope as agreed to by the performing organization and its stakeholders is delivered on time, within budget, and with sufficient levels of quality, as agreed to with the customer.
- Scope can be readily controlled, verified internally, and validated with the customer. Scope and quality management, along with stakeholder engagement and stakeholder expectation management, lead to customer satisfaction with project progress.
- The customer is satisfied with project deliverables.





## Scope Management Planning

Together, the requirements and scope management plans provide direction on how the project and product scope will be defined, managed, and controlled. The project charter, project life cycle, and development approach descriptions, and organizational process assets are all inputs to the process of planning scope management. The development approach influences how requirements will be elicited and how the scope statement will be developed.

### Scope Management Plan

The scope management plan, which is the primary artifact of planning scope management, is part of the project management plan. The project manager uses it to guide the building of the product until closing. It details how scope will be planned, executed, and controlled. It describes how to do the following:

- Achieve the overall project scope
- Create the WBS and WBS dictionary, or product backlog and stories
- Manage and control scope to the project management plan
- Obtain acceptance of deliverables

Each scope management plan must be tailored to the particular project, but it may cover topics that can be standardized for a company or for a particular type of project. Therefore, organizations often utilize templates, forms, and accepted standards for scope management. These are examples of organizational process assets.

### Requirements Management Plan

In addition to describing the methods the project manager intends to use to identify requirements, the requirements management plan should answer the following questions:

- Which requirements techniques will be used to analyze and document the requirements?
- Once I have as many requirements as I can gather, what will I do to analyze, prioritize, manage, and track changes?
- What should I include in the requirements traceability matrix? (Described later in this chapter.)

The requirements and scope management plans can be developed in stages or iterated during project planning. The first step is to plan how scope will be defined and who will be involved. Planning decisions will then become part of the scope management plan. Later planning efforts may result in scope being added so the process is iterative. For example, the completion of the Plan Risk Responses process means these risk responses are part of the project scope, causing a new iteration of the scope management plan, scope statement, and WBS, or the product roadmap and backlog, for agile.

### Agile Scope Planning

Project and product planning is more iterative on agile projects than it is on plan-driven projects. Specifically, iterative product planning happens first at a high level during product visioning. Then the team begins to decompose the product into a backlog containing stories, which may be decomposed further. Once decomposition happens, the product roadmap is created in more detail. Finally, the team plans for each iteration.



#### Agile Visioning

We discussed product visioning in the “PMP® Exam References in Context” chapter. This is about establishing a common understanding of what the product is, what it does, and creating a succinct way of describing it and the value it delivers. This is often referred to as an “elevator statement,” since it should be short enough to describe in a short elevator ride.

#### Agile Product Roadmap

A product roadmap alone is not the agile equivalent of a plan-driven scope management plan, but it is a visual representation of the product’s main components, broken into sequential product releases. It also acts as:

- A communication tool that provides stakeholders with a high-level view of the intended functionality of each release
- A high-level planning tool with the assumption that there will be changes to it
- A tool that allows you and the team to go back to confirm (and change) roadmap components over time

The roadmap and backlog work together like this to help the team plan the project:

- The roadmap shows how the product will grow by release.
- The backlog further breaks down the features in each release into smaller, more manageable pieces called stories.

The product roadmap and backlog influence each other, and changes to project priorities or requirements are reflected on both. Figure 7.2 shows one way a product roadmap might look for a software product that allows the consumer to manage appointments and bills on their health clinic's website. The "(P)" following some of the entries indicates a planned partial completion for that release, while "(C)" indicates a planned full feature completion. You will also notice at the bottom of column one that there are often "stretch" goals, which are agreed upon. This means that if the team finishes the stories for the release more quickly than anticipated they will work on the stretch goals, but the customer is already aware that it is a stretch and may not be completed or even started for that release.

Release 1	Release 2	Release 3
Comply with regulations (P)	Comply with regulations (C)	Patient can manage insurance and payments
Branding/style schemes	User interface design (UID)	Patient can view their data from other institutions (P)
Database integration (phase1)	Database integration (phase2)	
Site security	Patient can change personal data and preferences	
Manage web accounts (login, password, etc.) (P)	Patient can manage appointments	
Patient can view own medical data (P)	Patients with edit access could damage the database <u>RISK</u>	
Current website capacity may not be enough <u>RISK</u>		
System architecture <u>RISK</u>		
Conduct outreach marketing campaign (stretch) (P)		

**FIGURE 7.2** Patient client portal project roadmap example

### Agile Product Backlog

A product backlog is a single, visible master list of all functional and nonfunctional work identified for the project. In other words, a backlog is a list of work that needs to be done. Backlog stories have a description of each piece of functionality and are reviewed for risks by the team. They are prioritized by the product owner (as an integral team member). A story should also include the following information:

- The business benefit
- Definition of done for determining when it is complete
- Acceptance criteria for determining under what conditions (requirements) it will be accepted by the customer (like scope verification and validation)
- The stakeholder who requested it

Here is the principle behind the building and prioritizing of stories:

- The product owner organizes the backlog by priority from the top down.
- The highest value stories are always at the top, from which the development team pulls to build the product.



- While the product backlog contains all the formally recognized scope, low-priority items at the bottom of the backlog may never be developed if the cost or time to produce them is deemed greater than the value they would return. Because scope is emerging, this is sometimes not known until later in the project.
- Initial backlog prioritization happens early during release planning and then later again during iteration planning.
- The team builds stories for a current iteration while the product owner refines and reprioritizes the backlog for the next iteration. While doing this, the product owner also answers questions for the current iteration.

Figure 7.3 shows an example product backlog for the healthcare clinic patient website. Some additional functions of the backlog include:

- A single source of information about the project to aid in effective communication and provide a visible artifact of the project's scope and status.
- A tool for continually updating scope as the project progresses and new information becomes available.

#	Features	Stakeholders
P1	Manage appointments	Patients, administrators, practitioners
P2	Change personal data and preferences	Patients, administrators, practitioners
P3	View health information library	Patients, practitioners
P4	Outreach (marketing) campaigns	Patients, marketing
P5	Practitioner and patient communications	Patients, practitioners, marketing
P6	Regulation compliance	Patients, government
P7	View patient's own medical data from The Center	Patients, administrators, practitioners
P8	View patient's own medical data from other institutions	Patients, administrators, practitioners
P9	Manage web accounts (login, password, etc.)	Patients, administrators

**FIGURE 7.3** Partial product backlog for patient web page

### Iteration Planning

After release planning and the product backlog is created and prioritized, the team, including and importantly the product owner, decide which increments of the product will be built during the first iteration. Then as each iteration is successfully completed the product owner has already prepared the stories to be built for the next iteration, and so on throughout the project.

### Hybrid Project Planning

Parts of hybrid projects are plan-driven, and parts are planned and delivered using agile methods. For example, let's say there is a large, complex product that needs a complex variety of working features, like a control system for a new solar energy-based community. This system may need to be built using a plan-driven approach, while the rollout of the solar panels may be done iteratively using agile methods as new homes are completed in the community. The design, build, testing, and installation of the control system in the community building would be done using a WBS while the solar panel systems for the individual homes could use lists, like backlogs, that would be customized per customer.

Often, in mixed corporate environments, projects using a hybrid approach will have a WBS (or similar tool). Work will be broken down to a certain level and will be used to share information with the PMO or executive management, who may be accustomed to predictive approaches and traditional documentation. A backlog may then be used for organizing work with the team. The project manager acts as an interface between groups.

## Eliciting and Analyzing Requirements

Stakeholders can often describe a problem they have or an opportunity they want to take advantage of. Yet it is difficult or impossible for them to describe the solution. Business analysis is needed to help the project team elicit and analyze requirements before product and project scope can be fully defined. Requirements are the product features, or what stakeholders need from the product and from the project. The “Collect Requirements” process, as it was named in the Process Groups model, looks for all requirements, not just those related to the product. The process of eliciting and analyzing requirements is critical to project success, as a missed requirement could mean significant changes and conflict throughout the remainder of a project.

**Note:** The term “elicit and analyze” requirements better represents the magnitude and importance of this process. However, when referring to the historical name given to this process in the PMI Process Groups model, we will use “Collect Requirements.” You may see either term on the exam.

The objectives for which the project was initiated were originally the result of a Needs Assessment process that would have taken place to help establish the business case for the project. Using this information, all requirements should relate to achieving these objectives, as outlined and approved via the project charter. They may include requests about how the work is planned and managed.

**Example** A stakeholder could request that systems not be shut down during peak business hours to accommodate a project.

Requirements include the capabilities stakeholders need from the product, such as a software application that opens at a set time when it has updates for the user. But requirements can also be non-functional, in categories related to the following examples:

- **Quality** The component D must be able to withstand 200 pounds of pressure.
- **Business process** You must track and report the project’s expenses in this way.
- **Compliance** By law, we have to meet this safety standard.
- **Project management** We require risk management procedure X to be used.
- **Environmental consideration** Results of the full environmental impact study must be followed so the project has no negative impact on the environment.
- **Social need** The new transit line must have accommodations for people with disabilities who cannot drive cars.

Functional requirements are also often designed to answer specific questions, like the following:

- **Business requirements** Why was the project undertaken? What business need is the project intended to address?
- **Stakeholder requirements** What do stakeholders want to gain from the project?
- **Solution requirements** What does the product need to look like? What are its functional requirements (how the product should work) and nonfunctional requirements (what will make the product effective)?
- **Transition requirements** What types of handoff procedures or training are needed to transfer the product to the customer or organization?
- **Quality requirements** What quality measures does the product need to meet? What constitutes a successfully completed deliverable?
- **Technical requirements** How will the product be built? What are the product specifications?

### Process Groups Model

PG: Planning

Process: Collect Requirements

### ECO

Domain II

Task 8 Plan and manage scope

### PMBOK® Guide

Domain 2.4 Planning

Agile methods do not attempt to specify fully detailed requirements up front. Agile teams initially define requirements at a high level and then progressively refine them. This approach delays decisions on implementation details until the last responsible moment, helping to avoid or lessen the effect of change requests.



**Think About It.** The “Collect Requirements” process involves using the following inputs to create the requirements documentation and the requirements traceability matrix. These are artifacts needed in order to elicit and analyze requirements for a plan-driven project. Review them and think through how each may help to elicit and analyze requirements.

- **Project charter** The Collect Requirements process begins with descriptions of the high-level requirements in the charter. More detailed input from stakeholders is part of the reason for Collect Requirements.
- **Assumption log** This documents known stakeholder assumptions related to product and project requirements. Eliciting and analyzing requirements includes refining and adding to this list.
- **Stakeholder register** Created in initiating, this includes a list of stakeholders identified thus far, as well as their requirements and expectations.
- **Agreements** Buyers’ requirements are documented in contracts if the project includes procurements. Agreed-upon requirements included in letters of agreement internal to the organization are also a source of requirements.
- **Organizational process assets** Examples of these could be historical records and lessons learned. These may provide information about requirements from past, similar projects as well as information that may identify commonly overlooked areas of scope.
- **Stakeholder expectations** The Collect Requirements effort also includes eliciting stakeholders’ expectations—their beliefs or mental pictures about how the project will turn out—and translating those expectations into requirements as necessary. Not all expectations are requirements so this is an area requiring trust and effective stakeholder communication.

On large projects, there could be hundreds of stakeholders, and no single method of eliciting and analyzing requirements will work for all of them. Since missing a requirement can be costly, a concerted effort is made to find as many requirements as possible before work starts on a development phase.

## Methods for Eliciting and Analyzing Requirements

The project manager needs to tailor their method choices to the project and its stakeholders. The following methods are representative examples of those used to elicit and analyze requirements.

### Verbal Requirements Elicitation Methods

With conversational methods for eliciting and analyzing requirements, the exchange between the team—which may include a business analyst—results in written artifacts that are then used to define, plan, and manage project and product scope.

**Brainstorming** Many people think this is just a meeting where people discuss ideas, but it is more than that. The purpose of brainstorming is to get people to share ideas on a topic, but importantly, to build on each other’s ideas. It can be highly beneficial to include people with different perspectives or backgrounds. The participants may be internal or external to the project and/or the organization. Here’s how it works:

- One person mentions an idea to solve a problem or, in this case, elicit requirements and ultimately determine scope. No evaluation happens during the idea-generation period.
- The idea generates an idea from another participant, which leads to yet another idea, and so on.
- After the ideas have been captured, the group evaluates and ranks the ideas using the nominal group technique or multicriteria decision analysis (described later in this section).

**Interviews** You may also see the term “expert interview” on the exam. The project manager and/or a team member interview stakeholders to elicit their requirements for a specific element of the product or project work, or for the overall project. These interviews take place between two individuals or in group settings. They may also be conducted via email or phone or using virtual collaboration tools.



**Focus Groups** This technique elicits opinions and requirements for the product or project from stakeholders and subject matter experts. Usually selected from a specific demographic group of customers, focus group members discuss their ideas with each other. The conversation is directed by a moderator.

**Questionnaires and Surveys** These are typically used for large groups. Questions are crafted to specifically elicit requirements and expectations from respondents.

**Benchmarking** This looks at what the competition is doing. Benchmarking focuses on measuring an organization's performance against that of other organizations in the same industry. Limitations on this include that it can be time-consuming and costly. It may also inhibit creativity because the focus is on examining solutions that have been used rather than on innovation.

**Facilitation** This technique brings together stakeholders with different perspectives, such as product designers and end users, to talk about the product and, ultimately, define requirements. It uses a consensus approach, which achieves general agreement about a decision. Those who would prefer another option are willing to accept the decision supported by most members of the group. Facilitators sometimes use voting to help a group discuss various opinions.

**Voting** Voting is commonly used for group decision making. Soliciting input about requirements from stakeholders often results in conflicting requirements. A decision-making process might have the goal of unanimous agreement. Or, when there are conflicting opinions, groups may take a majority approach, taking the decision that more than half of its members support. If there is no majority opinion, the group may go with the decision that has the largest number of supporters. This is known as the plurality approach. PMs should be careful making decisions based on majority rules in case key stakeholders are in the minority.

Agile teams use voting all the time for group decision making but it may be more difficult to use with customers, in which case other methods should be used.

**Multi-criteria Decision Analysis** With this technique, stakeholders quantify requirements using a decision matrix based on factors such as expected risk levels, time estimates, and cost and benefit estimates.

**Nominal Group Technique** This technique can be (but is not always) done during the same meeting as brainstorming. It follows these steps:

- A question or issue is posed
- All participants write down their ideas privately
- Each participant shares their ideas
- The group discusses what has been shared
- The group ranks the ideas based on which are most useful in the given context

**Observation** This is a great way to learn about business processes and to get a feel for the work environment of stakeholders. This technique is useful for projects aiming to streamline a business process. It generally involves job shadowing—watching a potential user of the product at work, asking questions and, in some cases, participating in the work to help identify requirements.

**Prototypes** A prototype is a model of the proposed product that is presented to stakeholders for feedback. The prototype may be updated multiple times to incorporate stakeholders' feedback until the requirements have been solidified for the product.



**User Stories** Stakeholders may help develop user stories during facilitated discussion sessions. Stories describe functionality or features that stakeholders hope to see. They are often written in the following format:

As a <role>, I want <functionality/goal> so that <business benefit>

**FIGURE 7.4** Story format

**Example** “As a community organizer, I want the library to offer public meeting spaces so we have a place to gather and show community members the library’s benefits through neighborhood events.”

Other examples of facilitation sessions include the following:

- **Joint application design (JAD)** Used primarily in software development efforts, JAD sessions involve eliciting requirements and other input to enhance the process of developing the software.
- **Quality functional deployment (QFD)** Also referred to as the Voice of the Customer (VOC), this technique is generally used in manufacturing to elicit and prioritize customer requirements.

## Graphic Requirements Elicitation Methods

Different types of artifacts result from eliciting and gathering requirements, which will help to define, plan, and manage project scope. The following methods for eliciting and analyzing requirements result in graphic images, or different ways of representing project requirements.

**Mind Maps** This is a way of diagramming ideas or notes to help generate, classify, or record information. It branches out of a central core as shown in figure 7.5. Colors, pictures, and notations can be used to make the diagram more readable.

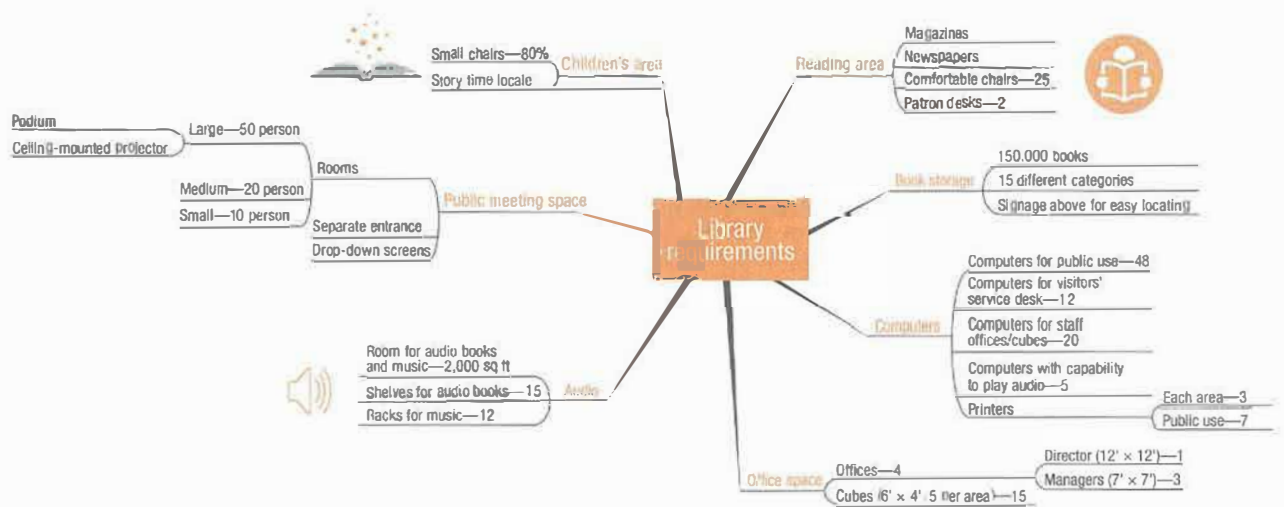


FIGURE 7.5 Mind map

**Context Diagrams** Also known as a context level data flow diagram, a context diagram is frequently used to define and model scope. It shows the boundaries of the product scope by highlighting the product and its interfaces with people, processes, or systems.

Figure 7.6 shows an example of a context diagram for the payroll system upgrade described in the project charter in the “Integration” chapter.

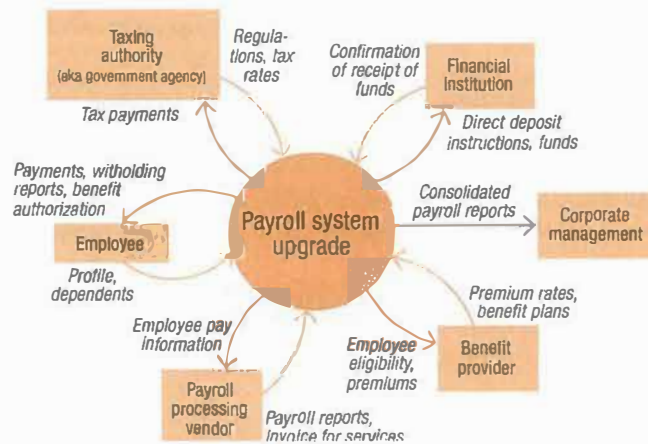
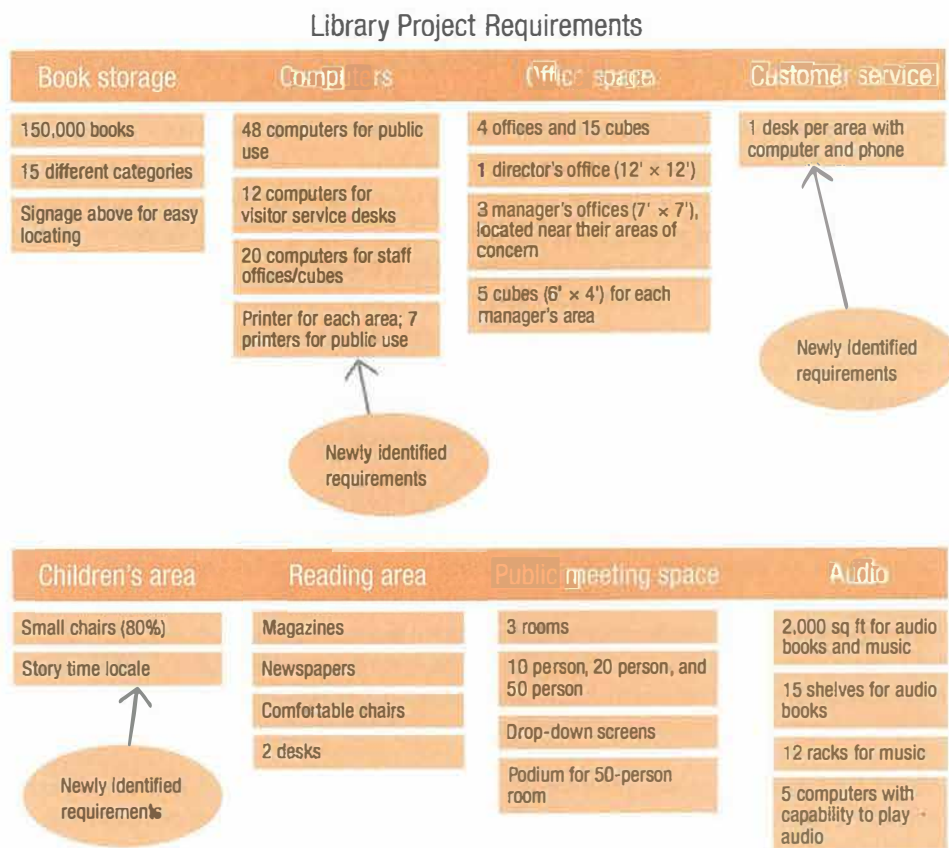


FIGURE 7.6 Context diagram



**Think About It.** Stop for a moment to think about how you might use the following two methods for eliciting and analyzing requirements. You may initially create a context diagram with the help of stakeholders, to get overall understanding of the product and who might use it. You may then use a combination of surveys, observation, and facilitated discussions with stakeholders to elicit their requirements. Then, as part of analyzing the requirements you have elicited you might sort them into an affinity diagram so that similar items are grouped together for further analysis into how they relate to the overall product.

**Affinity Diagrams** This technique groups requirements (generated from other gathering methods) by similarities. This sorting makes it easier to see additional areas of scope that have not been identified. Figure 7.7 shows an example of an affinity diagram.



**FIGURE 7.7** Affinity diagram



## Artifacts of Eliciting, Analyzing, and Balancing Requirements

Important aspects of eliciting and analyzing requirements are balancing them against one another, resolving conflicts where there are competing requirements, and, finally, completing the requirements artifacts, so they will help you with the next process, which is Define Scope.

### Balancing Requirements

Part of balancing stakeholder requirements involves making sure the requirements can be met within the project objectives. If they cannot, then the project manager needs to look for options to adjust the competing demands of scope, time, cost, quality, resources, risk, and customer satisfaction.

This is never easy or fast. It can become impossible if there aren't clear project objectives. Do you try to get as close to final requirements as possible when managing projects? Are your requirements ranked by order of importance? If not, think about how such actions could improve your projects. When you take the exam, assume that every effort has been made by the project manager to uncover all requirements to the degree that can be known, and that those requirements are ranked by order of importance.

Agile and hybrid approaches are often used when the exact requirements are unknown.

**Example** You may be building something your organization has not done before, such as creating a customer (or patient) self-service portal that allows each customer to manage their own account. In this case:

- You may not know what the most popular functions will be or the exact extent of the scope.
- You may allow customers to change their own name and address fields and enrolled services, but what about deleting their accounts?
- Do all the key stakeholders agree about what features should and should not be included in a self-service portal?

Additional requirements are often uncovered after some use of the product or service as well.

- After launching the self-service portal, you might learn that a large percent of customers access it on a mobile device, but the portal was optimized for a PC.
- You might also discover a competitor's portal features a "refer-a-friend" option that offers rewards, and you may want to create a similar program.

A project like this, where the scope is not well defined, may be best completed using an agile or hybrid project. You could, for example, use questionnaires and surveys to find out what the most important and first features of the product should be. You could create committed focus groups so that as the first release is rolled out you have customers willing to use it and help you further develop the features to come next, and so on, until the product is mature and periodic upgrades are all that are needed.



**Think About It.** Often, scope and priorities change. The longer the project, the more likely the scope will change because of changes in the market, technology, or organization. Take a moment to think about the patient portal example in terms of the ECO's Business Environment domain. Here are some examples:

- **Plan and manage project compliance** (task 1) How often do the regulations change? What type of SME (subject matter expert) is needed to ensure compliance, and where will those SMEs come from? Outside the United States, what are the compliance requirements and how will you ensure your site complies with them?
- **Support organizational change** (task 4) How will you ensure that everyone within the clinic understands the product vision, is trained on using it but also embraces the changes to come within the organization and for the customer (patient). This type of supporting organizational change is so significant and often so big that it could be done as a sub-project or a "release" of its own.



### Resolving Conflicting Requirements

It is often difficult to prioritize conflicting requirements.

**Example** Consider the following examples from an organization that is embarking on a project to improve their product development processes:

- What if the engineering department wants the project to focus on decreasing defects while the accounting department wants the project to focus on lowering costs? Can both needs be met?
- What if the engineering department is the primary stakeholder or even the sponsor of the project? Should that department's needs outweigh the needs of the accounting department?
- What if the needs of the engineering department negatively impact the accounting department?

Some issues cannot be resolved by the project manager and team. These require sponsor or other management intervention. However, there are some standard guidelines for balancing competing requirements. For the exam, keep in mind that competing requirements can be resolved by accepting those that best comply with the:

- Business case
- Project charter
- Scope statement
- Known project constraints

Here are additional considerations:

- Reject a stakeholder's request to do or add something that is not related to the reason the project was initiated. It cannot deliver project benefits if it is not related to the project charter.
- If a requirement is related to the reason the project was initiated but does not fall within the project charter, this request should also be rejected. The project manager could encourage submission of a new project request instead.
- Suggested changes to the project charter must be brought to the sponsor for approval. Typically a project charter does not change beyond the initiation and visioning stages of the project.

When considering constraints, if the most important constraint is schedule, then:

- New requirements that would delay the schedule will not likely be accepted. On an agile project, of course, they could be added to the backlog and other requirements deprioritized to another project. An agile project backlog does not have to be completed at the end of a project because some features could be deferred to another, future project.
- New requirements that compress the schedule or at least do not delay the schedule (without serious impact to other project constraints) will likely be accepted.

Whether a project is plan-driven, agile, or hybrid, requests that do not fall within these guidelines could become part of a future project instead.

## 7.1 Exercise

This exercise describes some of the key actions involved in balancing stakeholder requirements. It goes beyond the Collect Requirements process and looks at this effort throughout the project. Spend time thinking about balancing requirements. This exercise will help you determine whether you really understand the process.

In your Exercise Notebook, create a table like the one below (you do not need to write down every action, simply write down the number). Read through each action and place a checkmark in the “Know” column if you understand the action described. Put a checkmark in the “Do” if you actually apply the action in the real world. After you’ve gone through the list, make sure you return to the actions without two checkmarks and spend time working through them in a way that makes them real to you so you can answer related questions on the exam.

Action	Know	Do
1. Identify all stakeholders; understand their needs, wants, assumptions, and expectations for the project.		
2. Get requirements as clear and complete as appropriate for the selected development approach before starting project work.		
3. Use information about stakeholders and their requirements to resolve competing requirements while work is being done on the project.		
4. Look for competing interests during project planning; don’t wait for competing interests to show up during execution.		
5. Look for possible options to resolve competing interests and alternative ways of completing project activities. This may involve using techniques such as brainstorming, schedule compression, reestimating, and other practices.		
6. Resolve competing requirements from stakeholders based on how the requirements affect the project.		
7. Give priority to the customer. (For the exam, know that if any needs conflict with those of the customer, the customer’s needs normally take precedence.)		
8. Use quality management to support the project’s satisfaction of the problems or opportunities for which it was undertaken.		
9. Deal with problems and conflicts as soon as they arise through the use of consensus building, problem-solving, and conflict management techniques.		
10. Say no to some of the competing interests. (For the exam, assume the project manager has the authority to say no when necessary to protect the project.)		
11. Fix the project when the project metrics start to deviate from the requirements, rather than changing the requirements to meet the results of the project.		
12. Work toward fair resolutions to disputes—solutions that consider the interests of all stakeholders as well as the needs of the project.		
13. Hold meetings, interviews, and discussions to facilitate the resolution of competing requirements.		
14. Call on management to help resolve competing interests when the project manager and team cannot come up with a fair and equitable solution.		
15. Use negotiation techniques to resolve conflicts between stakeholders.		
16. Plan and implement effective communication.		
17. Gather, assess, and integrate information into the project.		



### Verifying Requirements

It is important for plan-driven and agile projects alike to verify requirements at every opportunity. This often entails meeting with the customer to clarify requirements that are not clear and discuss requirements that have already been elicited in order to make sure they are well understood. There should be a common understanding about requirements between the customer and team. Often prototypes are useful to show the customer and ensure the requirements are well understood before building an increment of the product.

**Iteration Reviews (Post-iteration Product Demos)** Projects that use agile and hybrid approaches frequently demonstrate completed increments. Sometimes stakeholders are not interested in early increments where there is not yet much visible functionality. But this is exactly when teams most want feedback. Getting stakeholders involved in early reviews is important because it stimulates their ability to see and articulate what their requirements really are. The project manager and the team must explain how important this early feedback is to get the design and features right before change becomes more costly.



Project managers on agile and hybrid projects should explain the cost of change (see “Stakeholders” chapter) to stakeholders. Teams want to discuss changes when the product design is still in development and the cost of change is relatively low. This requires the team to have courage to demonstrate incomplete solutions that may face criticism and the business to have trust and imagination into how the system may look in order to provide feedback as soon as possible.

### Requirements Documentation and Other Artifacts

After requirements have been elicited and analyzed, verified, and prioritized, documentation can be completed. Requirements documentation on predictive projects is typically more formal than on adaptive projects which use lightweight techniques like hand-drawn diagrams. Imagine eliciting requirements from hundreds of people. Can you see how documenting those requirements would be useful? This documentation is an output of the Collect Requirements process and helps to ensure all requirements are clear and unambiguous.

**Acceptance Criteria** Requirements documentation can contain many types of information, but one thing that must be included is acceptance criteria. To avoid having requirements that could easily be misunderstood, a great question to ask stakeholders is, “How will we know if the work we do will meet this requirement?” Not only is this a good way to make sure you understand the stakeholder’s requirement, but it also helps to ensure the work being done will be acceptable.

**Definition of Done** Most often associated with agile projects, this is beneficial on any type of project. Teams specify a “definition of done” for each product component at the user story and the release levels, as well as at the final product deliverable level. For example, imagine a case study where we are building a house. Because funding is only going to be available at different intervals, the house has to be built using an agile life cycle. The following examples describe when deliverables for the house are “done,” and the final product is the completed house:



- **User story level** (The story is “Concrete Curing completed”) “Curing completed” requires first that the foundation is laid, and the concrete has been poured. The “Concrete Curing Completed” is done after 7 days which is sufficient to allow the weight of walking on it (24-48 hours) but also the weight of construction vehicles (7 days).
- **Release level** (“Foundation complete”) “Foundation complete” is “done” when the foundation is laid, concrete curing completed, tests completed, inspection completed, shown to homebuilder, response to homebuilder feedback is completed, and homebuilder has approved/signed off on it.
- **Final product deliverable** (The “dream house” project is completed!) “Dream house” is “done” when all high- and medium-level priorities are complete according to their individual definitions of done, all inspections and inspection sign-offs are complete. The buyer has moved in and has successfully used all mechanicals in everyday usage for two months. They have completed a customer satisfaction survey giving at least a 4 on a 1-5 scale. (Note that low-level priority items for a home build might include finished landscaping or a planned finished basement. The buyer may take ownership with these items remaining on a backlog.)

**Relationship to Validate Scope** Requirements must be described in such a way that associated deliverables can be tested or measured for the Validate Scope process to confirm that the deliverables are acceptable. The level of documentation detail is iterated until each requirement satisfies the criteria of being clear, complete, and measurable, and acceptance criteria are established.

**Requirements Traceability Matrix** Have you ever worked on a project in which some requirements got lost in the details? It can be difficult to remember where a requirement came from, what its significance is to the project, or what other requirements it is related to. Losing track of requirement details can result in a project objective being missed. The requirements traceability matrix is a form of requirements documentation that helps link requirements to the objectives and to other requirements to ensure the strategic goals are accomplished. The matrix is used throughout the project in analyzing proposed changes to project or product scope. An example of a requirements traceability matrix is shown in figure 7.8.

Information like requirement identification numbers, the source of each requirement, who is assigned to manage the requirement, and the status of the requirement should be documented in the requirements traceability matrix. For large projects, however, including all this information in the matrix would make it cumbersome and difficult to use. Another option is to store this data in a separate repository, preserving the matrix as an easy-to-reference tool. For the exam, simply understand that the requirements traceability matrix links requirements to objectives and/or other requirements, and that the requirements attributes, such as identification numbers, source, and status, also need to be documented.

Assigning responsibility for certain requirements management is similar to the concept of assigning risk owners, described in the “Risks and Issues” chapter. Assigning team members to manage certain requirements helps to ensure the objectives are met and also helps free up the project manager’s time. Requirement ownership is another type of work team members may do on a project in addition to their work to produce the product. If a business analyst is on the project team, they would manage requirements.

Objectives	Reading area		Book storage		Public meeting space		Children's area		Audio		Office space		Computers									
	Magazines	Newspapers	Comfortable chairs—25	Patron desks—2	150,000 books	15 different categories	Signage above for easy locating	Rooms	Separate entrance	Drop-down screens	Small chairs—80%	Story time locale	Room for audio books and music	Shelves for audio books—15	Racks for music—12	Offices—4	Cubes—15	For public use—48	For visitor service desk—12	For staff—20	With audio capability—5	Printers
Improve access to job resources by 20%.		X		X				X		X								X	X			X
Improve local children's reading levels by two grade levels in one year.			X		X	X						X						X				
Provide a pleasant place for community members to meet.	X	X	X					X	X		X		X	X	X							
Replace the existing library by end of next quarter.	X	X	X		X	X	X					X		X		X	X	X		X	X	X

**FIGURE 7.8** Requirements traceability matrix

## Define Scope

The Define Scope process is concerned with what specifically is and is not included in the project and its deliverables. This process uses the requirements documentation just discussed as resulting from the “Collect Requirements” process. Other artifacts to work with are the project charter, scope management plan, assumption log, and the risk register.

### Process Groups Model

PG: Planning  
Process: Define Scope

### ECO

Domain II  
Task 8 Plan and manage scope

### PMBOK® Guide

Domain 2.4 Planning

## Predictive Project Management and Scope Definition

This is how scope definition plays out in predictive environments:

- Everything known at a high level about scope was documented in the project charter.
- Many more details about scope are uncovered and documented during the Collect Requirements process. At this point the requirements determination is sufficient to finalize the scope definition for the project management plan.
- Once the project management plan (with this definition in it) has been approved there may be changes to it but they will be subject to the Integrated Change Control process.

## Adaptive Scope Definition

How would you define scope in adaptive environments? Let’s look at this process from an agile perspective. A good reason to use agile is that scope is emerging, so scope will be relatively flexible.



### Agile Requirements and Scope Definition

In adaptive environments, requirements and scope definitions are emerging through much of the project. On agile projects scope is defined with progressive elaboration; first at the chartering and visioning levels in Initiation, and then with the creation of a high-level backlog and a release plan. Agile teams then decompose product feature requirements from the high-level backlog by progressively elaborating feature requirement details and “slicing,” or decomposing, high-level stories into smaller, more manageable pieces of work, much like creating and decomposing the work packages of a WBS in a predictive environment. Decomposition is discussed in the *Agile Scope Decomposition* section of this chapter.

- **Visioning and chartering** The value and benefits of the product of the project is succinctly described during visioning (in feasibility—not shown here). In chartering, project objectives are documented.
- **Backlog** The high-level list of features is created.
- **Release plan** This is completed with the previous two artifacts. The plan (or release map) will show at least the known functionality for the first release but could show further releases.
- **Detailed requirements** These are collected later as features and stories from the backlog are decomposed and the team prepares to build prioritized features for the first release. Story writing workshops are held to “slice,” or decompose, those stories to the most manageable size and complexity.

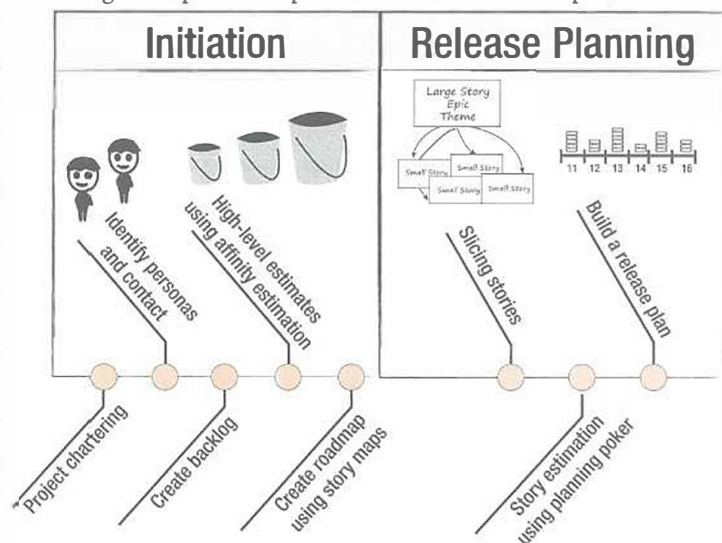


FIGURE 7.9 Defining scope in agile



### Flexible Scope Definition in Favor of Time and Cost

Timeboxes allow agile teams to define and manage a flexible scope to achieve the highest priority, best-quality product within a fixed cost and a fixed timeframe. You can think of scope on an agile project then to be very flexible while time and cost are more fixed. The highest priority items are achieved within the time given for the project.

### Product Analysis As a Scope-defining Method

As noted at the beginning of this section, part of defining scope is determining what the deliverables of the project are. Product analysis is a method of analyzing the objectives and description of the product as stated by the customer or sponsor. That information is then used to define tangible deliverables. The work of product analysis may entail analyzing the product description and stated requirements, or using techniques such as systems engineering, value analysis, or value engineering.

Product analysis allows the project manager to make sure the product and project scope are understood and accurate. For the exam, realize you may need to determine and define deliverables as part of the project, rather than receiving a complete list from the customer.

## Artifacts of Defining Scope

Project artifacts resulting from the Define Scope process for a plan-driven project include the project scope statement and updates to other project artifacts like the requirements documentation, a requirements traceability matrix if one will be used, and updates to the stakeholder register and the assumption log.

### Project Scope Statement

This is the primary artifact of Define Scope. This document in effect says, “Here is what we will do on this project,” or “Here is the approved project and product or service scope for this project.” The project manager and the team will have had many discussions with stakeholders. Many things that are not in the project will have been discussed and not accepted because they were not in the charter and did not belong to the project. So in the project scope statement, the project manager must also be sure to identify what is not in the project. They should also clarify areas where they have learned there are elements of the scope that could be easily misunderstood.

The project scope statement in predictive environments typically includes the following:

- Product scope
- Project scope (including descriptions of project management components)
- List of product deliverables
- Acceptance criteria
- What is not part of the project
- Assumptions and constraints

In agile there may or may not be a scope definition in the form of a scope statement. There will definitely be components of scope as described in the Adaptive Scope Definition section of this chapter. Additionally, agile teams decompose large or complex stories into smaller, more manageable stories. These smaller “more manageable stories” are analogous to work packages of a WBS on a plan-driven project. What this means is they’ve been broken down so that they can be more easily estimated for time, cost, and other resource needs and assigned to team members to be built. Completed stories prioritized to be part of a release are then integrated together to become a working increment of the product that can be released to the customer.



## Create WBS (Decompose Scope)

In planning for predictive practices, scope decomposition consists of creating a WBS and WBS dictionary. The scope definition consisting of the scope statement, WBS, and WBS dictionary make up the project's scope baseline, so there is no questioning the need for all three of these components.

The following sections about the work breakdown structure (WBS), WBS dictionary, and the scope baseline are largely from the Process Groups model's perspective. We will discuss agile scope decomposition following these sections.

### Process Groups Model

PG: Planning

Process: Create WBS

### ECO

Domain II

Task 8 Plan and manage scope

### PMBOK® Guide

Domain 2.4 Planning

## The Work Breakdown Structure (WBS)

What is a WBS? Understanding and using this tool is essential for successful projects using a traditional approach, and for passing the exam. Start by testing your current understanding.

### 7.2 Exercise

What does a WBS contain and what is its value as part of the scope baseline? Write the answer in your Exercise Notebook.

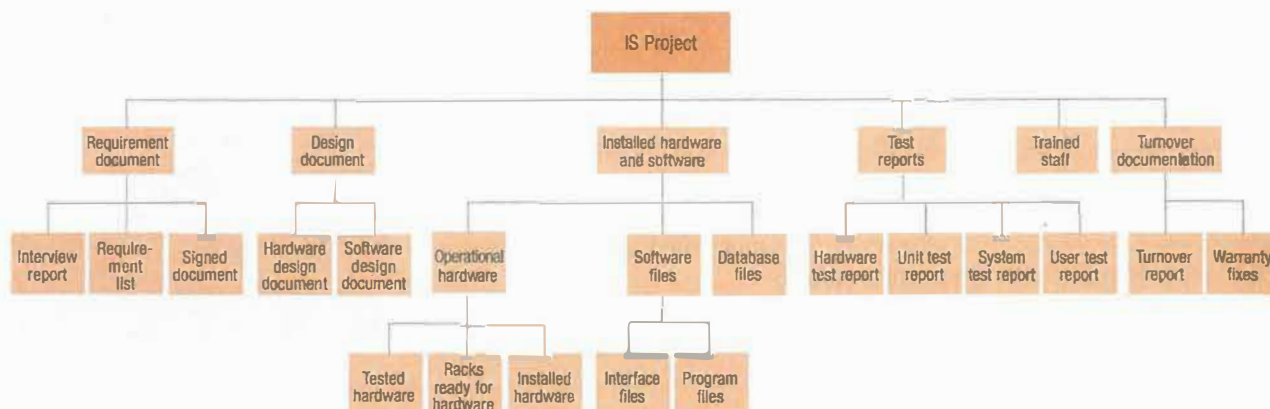
### Answer

The WBS is a visual, organizational tool (like an information radiator!) showing all the scope on a project, broken down into manageable deliverables called work packages. It helps ensure that no deliverables are missed. It is also a communication tool since it gives an image of what is included in the project.

Here are a few additional answers that may further define a WBS and its value to the project.

- The construction of a WBS graphically provides a structured vision for a project and helps to ensure that nothing, including deliverables, is forgotten.
- A WBS is created with input from the team and stakeholders. Involving the team and stakeholders helps gain buy-in, and increased buy-in leads to improved performance.
- The process of creating a WBS allows the team to go through a project in their minds and thus improves project plans. The execution of a project is typically easier and less risky as a result.
- Being involved in the creation of a WBS helps people better understand a project. It also makes a project seem more achievable.
- A WBS shows a complete hierarchy of a project, making it easier to see how one deliverable relates to another.

Review the WBS example in figure 7.10.



**FIGURE 7.10** A (summary level) WBS for a hardware/software creation and installation project

Decomposition, of course, needs to be tailored to the project. Typically the project name goes at the top of a WBS and the next level is the development life cycle. Subsequent levels break the project into deliverables, which are then broken down in succession until decomposition gets to the work package level (described next).

Did you know that a WBS allows you to break down a seemingly overwhelming project into pieces you can plan, organize, manage, and control? The creation of a WBS is an effort to decompose deliverables into the smaller component deliverables (work packages). Decomposition can be done using a top-down approach (starting with the high-level pieces of a project), a bottom-up approach (starting at the work package level), or by following organizational and industry guidelines or templates.

For the exam, know that on a WBS, work refers not to an activity, but to the work products that result from an activity or group of activities. Work packages are things (deliverables, product) rather than actions (activities). The complete product scope as well as the project scope (including project management activities) are included.

### WBS Guidelines

Every WBS is unique, and every project manager will create a WBS in their own way. For the exam, here are some guidelines that every project manager should follow:

- The project manager creates the WBS using input from the team and other stakeholders.
- Each level of a WBS is a breakdown of the previous level.
- The entire project should be included in the highest levels of the WBS. Not every level needs to be broken down further but those resulting in work packages must be broken down to the work package level.
- A WBS includes only project deliverables that are requirements. Deliverables not included in the WBS are not part of the project.

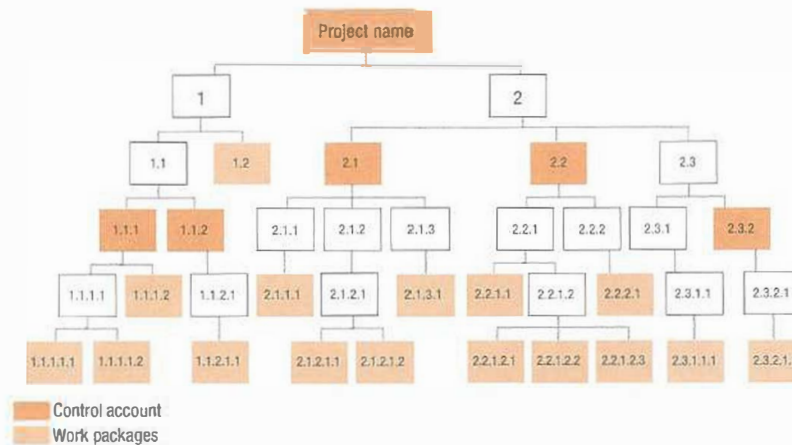
During planning, the project management team and subject matter experts break down the scope definition until the work package levels are reached. This occurs when the deliverables:

- Can be realistically and confidently estimated (including the activities, duration, and cost associated with them)
- Can be logically assigned to a distinct resource or resources
- Can be completed relatively quickly
- Can be completed without interruption and without the need for more information
- May be outsourced with minimal or no disruption to the internal team.

The levels in the WBS are often numbered for ease of reference. WBS software does this automatically and there are different numbering systems you can use.

Figure 7.11 provides an example.

On the exam you may see the term “planning package” or “control account,” as seen in the figure.



**FIGURE 7.11** Sample WBS numbering system

Control accounts, which may include one or more planning packages, allow you to collectively manage and control costs, schedule, and scope at a higher level than the work package. Each work package in the WBS is assigned to only one control account.

### The WBS and Schedule Planning

As planning progresses the team breaks down the work packages from the WBS into the schedule activities that are required to produce the work packages. The activities and their data are typically entered into a Gantt chart of traditional project management software. The data can help to create the project schedule. Often project management software can also generate a network diagram from the Gantt chart data. In the following list you can see from some of the basic activity data how planning scope is related to planning schedule, cost, resources, and other constraints:

- Name (of work package)
- Duration
- Dependencies (what must be done before this work package?)
- Start date
- Finish date (typically driven by duration)
- Resource assigned

### Scope Statement, WBS, and WBS dictionary

The team uses these three plan components to help to define the activities required to produce the deliverables. The WBS dictionary is described in the next section of this chapter, and the Define Activities process is described in the “Schedule” chapter.

### How Work Packages Are Defined

Just how project scope is broken down into work packages is tailored to the project. Historically the guidelines have been that on small projects a WBS is often broken down into work packages that take between 4 and 40 hours to complete. Medium-sized projects may start with work packages of 8 to 80 hours of work, while large projects may start with packages of 300 hours of work. While we include them here in case you run into them on the exam, these historical examples are not as likely to appear in questions as they once were. This practice has decreased in favor of a subjective-level decision by the team as to “what makes sense” in terms of being able to estimate, resource, and build a work package. Since the project is planned iteratively regardless of the approach, work package definitions can be iteratively refined.

### The WBS As Organizational Process Asset

If your company works on many similar projects, the WBS from one project may be used as the basis for another. Expect for exam purposes that the PMO collects and shares WBS examples, encouraging the creation of templates.

Do you really understand the WBS and its importance to planning and managing scope in predictive environments? Try the next exercise. If you miss many of the answers, then this is a gap area for continued review before the exam.



### 7:3 Exercise

Test yourself! What are the benefits of a WBS? Write the answers in your Exercise Notebook.

### Answer

The following are benefits of using a WBS:

- Gives a picture of the entire project's scope
- Helps people better understand the project
- Provides the team with an understanding of how deliverables fit into the overall plan
- Gives team members an indication of the impact of their work on the project as a whole
- Facilitates communication and cooperation for the project team and other stakeholders
- Helps manage stakeholder expectations regarding deliverables
- Helps identify risks
- Helps prevent work from slipping through the cracks
- Helps prevent unnecessary changes
- Focuses the team's experience on what needs to be done, resulting in increased quality and a project that is easier to manage
- Provides a basis for estimating resources, costs, and schedules
- Provides proof of the need for resources, funds, and schedules
- Helps with planning control efforts
- Helps in establishing deliverable acceptance criteria
- Gets team buy-in and facilitates team building

A WBS is a foundational component of project management. Almost everything that occurs in planning revolves around the WBS.

We have already mentioned the relationship of scope planning for cost, schedule, and resources. Figure 7.12 illustrates how many other project planning components rely on the WBS. Following are some examples of how the WBS facilitates planning with project constraints:

- During project selection and during initiation, costs and the schedule are estimated at a very high level and for the project as a whole. During planning, though, costs and the schedule are estimated at the work package or activity level.
- A WBS can help a project manager and team identify more risks by examining a project at the work package level.
- Resource planning and management are aided by the WBS. Work packages are assigned to resources by work package and activity.



**FIGURE 7.12** Much planning revolves around the WBS

Think ahead for a moment to the project control aspect of having a WBS. The following exercise will help you review how the WBS is used beyond planning to also control the project as the team is building the product.

## 7.4 Exercise

What do you do with a WBS during executing and controlling the project? The WBS is created and used in planning but the exam also tests your knowledge of how it is used throughout the rest of the project. So, take some time to really think about this question. Write the answers in your Exercise Notebook.

### Answer

When completed, the WBS can be used any time the project scope needs to be evaluated. You may have thought of other things, but some examples follow.

- **Scope-related change requests** A project manager can use the WBS, along with the project scope statement, to determine if a change request is within the approved project scope.
- **Impacts of change** The project manager and team can use the WBS as part of the integrated change control process to evaluate impacts of requested changes to project scope.
- **Controlling scope creep** Project managers can control scope creep by using the WBS to reinforce what work is to be done. ("Scope creep" refers to scope increasing without appropriate change control processes.)
- **Communications** The WBS is a communications tool when discussing the project among the team or with other stakeholders (e.g., the sponsor, the customer).
- **Team orientation** The WBS can facilitate new team members understanding their project roles.

### TRICKS OF THE TRADE

Now, would you like to get more exam questions right? First, know that the exam may use the term "deconstruction" as well as "decomposition." Both terms mean the same thing. Second, many people confuse the terms "WBS" and "decomposition." They are related but there is a distinction. The best way to think of decomposition is that decomposition is what you are doing, and a WBS is the method to do it, and it is also the artifact that results from the effort. In other words, you decompose a project and manage its scope and other constraints using a WBS.

## WBS Dictionary

Think about how a work package is identified in a WBS. It is usually described using only one or two words. But assigning a deliverable with such a brief description to a team member allows for too much variation (which itself could cause scope creep). The WBS dictionary is the documentation providing details needed to build each work package. It also lists acceptance criteria for each deliverable, ensuring the resulting work matches what is needed. The project manager and team can use a WBS dictionary to further understand the work that needs to be done, and to prevent scope creep before work even starts rather than dealing with scope creep while the work is being done.

The WBS dictionary is an output of the Create WBS process. It may be used as part of a work authorization system, which informs team members when their work package is going to start. You can also use it to clarify a stakeholder's understanding of effort needed for a work package. Figure 7.13 is an example of a WBS dictionary. A WBS dictionary can include descriptions of:

- Schedule milestones
- Acceptance criteria
- Durations
- Interdependencies
- Other work package information

Control Account ID #	Work Package Name/Number	Date of Update	Responsible Organization/Individual
Work Package Deliverable Description:			
Work Involved to Produce Deliverable:			
Acceptance Criteria (How to know if the deliverable/work is acceptable):			
Assumptions and Constraints:			
Quality Metrics:			
Technical Source Document:			
Risks:			
Resources Assigned:			
Duration:			
Schedule Milestones:			
Cost:			
Due Date:			
Intependencies (before this work package):			
Interdependencies (after this work package):			
Approved by:		Date:	

FIGURE 7.13 WBS dictionary

Note: Some of the entries in a WBS dictionary, such as durations and interdependencies, may be filled in during planning iterations rather than when the WBS and WBS Dictionary are first drafted. Interdependencies, for example, are best defined and understood as a result of doing a network diagram, which is part of schedule planning.

## Scope Baseline

In predictive environments the scope baseline is a set of artifacts that make up part of the project management plan and includes the project scope statement, the WBS, and WBS dictionary.

The scope baseline is approved at the end of planning and before the work of building the product begins. Then, as the work on the project is being done, the project manager reviews how the project is progressing and compares that data to the baseline by answering the following questions:

- What scope has been completed on the project?
- Does it match what is defined in the WBS, WBS dictionary, and project scope statement?



### Changes to the Scope Baseline

If scope is needed that is not in the baseline:

- A change has to be formally approved through the Integrated Change Control process.
- A new item (or items) needs to be added to the WBS, WBS dictionary, and project scope statement to reflect the change.
- The updated documentation becomes the project's new scope baseline.
- Other artifact components that are affected by the scope change need to be updated, including (most commonly) requirements documentation and the assumption log.

### Performance Measurement

Measurements of project success include whether the project has met all the requirements, including the scope baseline. It is essential to use these tools of project management in the real world. Aside from their use in achieving project success, you need to feel comfortable with them for the exam.

### Agile Scope Decomposition

Now let's talk about decomposing project scope from an agile perspective. This is worth repeating from the Adaptive Scope Definition section of this chapter:



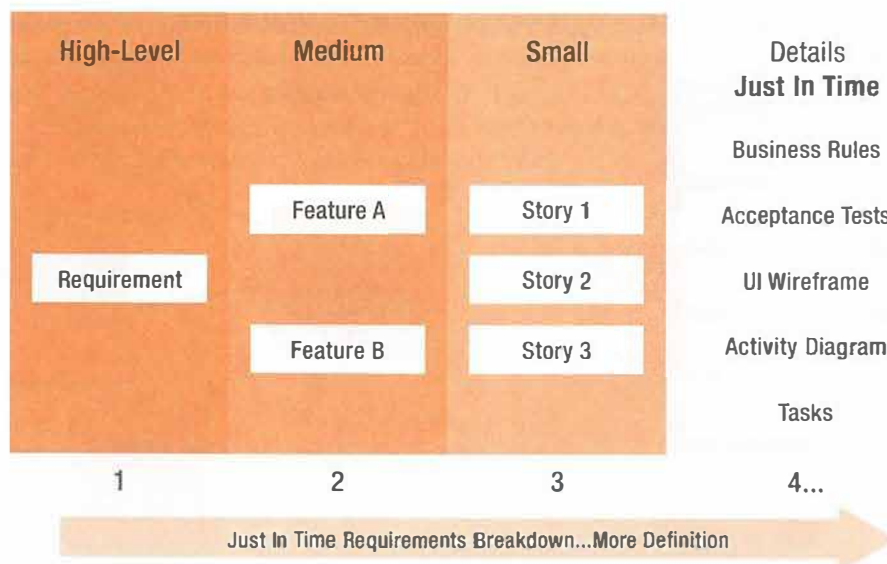
- In agile, product scope is defined at a high-level during project visioning and chartering.
- After this, a high-level product backlog is created, and release planning is started.
- From that high-level product backlog, product features are decomposed into smaller stories that can be estimated, assigned resources and built.

Notice that this is in essence the principle of progressive elaboration. Early on the project manager and the team gather high-level scope definition details. From there product features are progressively decomposed into these smaller stories through story writing workshops and "slicing" (decomposing) stories.

### Requirements Identification and Decomposition Progression



**Think about it.** The following is agile scope decomposition, illustrated. Figure 7.14 shows levels of agile requirements. Following the figure, the sections of the illustration (numbered for convenience) are explained in greater detail. Then, walk through an agile project in your mind, using the health clinic client portal case as an example.



**FIGURE 7.14** Levels of agile requirements



1. **High-level** Requirements or objectives are identified at the beginning of the project. Although agile practitioners accept that scope is emerging, they still make the effort to uncover as many requirements as possible from the start. It is unlikely you will see the word *epic* on the exam but in case you do, know that some agile practitioners use it to describe the biggest of requirements (or stories). Epics are too large and complicated to build without decomposition.
2. **Medium** *Features* are created from large and complex, high-level requirements. The level of decomposition here is called “medium” in the figure but notice that we are just using relative terms. There is no exact definition of “high-level” versus “medium,” etc.
3. **Small** Medium-level requirements are broken into smaller *stories*.
4. **Details** Each story needs to be broken further by various types of requirements, some of which are depicted in figure 7.14. The breakdown is described as *just in time* because agile teams wait for the *last responsible moment* to make sure they have all the details to build a story. The last responsible moment is the moment at which story decomposition has reached its logical conclusion and the most information is available about what needs to be done to build a story.

### Agile Decomposition: Health Clinic Client Portal

First and foremost, agile product feature prioritization and decomposition is value-based, always done from the standpoint of what stakeholders find most valuable.

- **Release map scope decomposition level** Review figure 7.2 on page 159 of the patient client portal project roadmap example. This will make it easier to walk through the project in your mind. The roadmap for the patient client portal project is shown in three releases. These are the high-level requirements (or epics).
- **Product backlog decomposition level** Review figure 7.3 on page 160: This is the product backlog. Listed in it is likely to be a combination of high- to medium-level features, which must be decomposed further into smaller stories. The “manage web accounts” set of features shown in the product roadmap for Release 1 could be comprised of P1 (manage appointments), P2 (change personal data and preferences), and maybe other features in this backlog.
- **Story decomposition level** A given feature is usually sliced (decomposed) into two or more stories; as this is done more details about the individual stories emerge. Stories can be compound, complex, or both. Figures 7.15 and 7.16 gives examples of each in the patient portal case study.
  - ✓ A compound story has multiple independent stories within it
  - ✓ A complex story is a big story that has to be broken down to decrease its complexity
- **Detailed decomposition level** At this level the team needs to have detailed story information established, like definitions for business rules, unit tests, acceptance tests, and tasks to build the story.

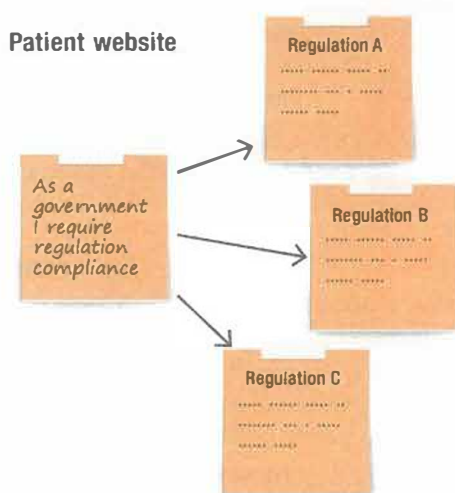


FIGURE 7.15 Slicing a compound story

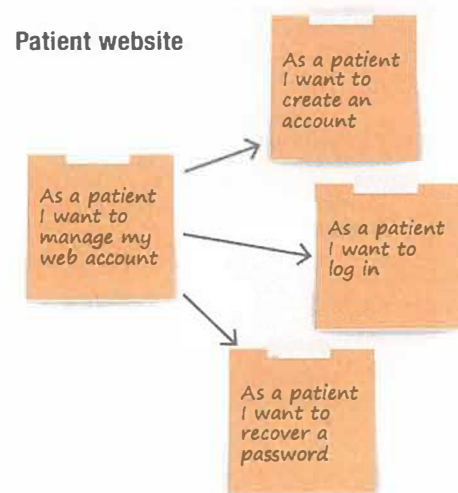


FIGURE 7.16 Slicing a complex story

### Other Agile Scope Decomposition Opportunities

In addition to product and project visioning, product roadmap, release, and iteration planning opportunities for scope decomposition on agile projects, agile teams are aware of the fact that throughout the project, opportunities emerge for better and different ways of breaking down requirements so the related stories are easier to build. Here are a few examples:

- **Example from the daily standup** At a daily standup meeting for the patient client portal project, a team member mentions they are having problems with regulation C because its technical requirements in the program interfere with the technical requirements of regulation A. After the meeting, the project manager gets the team member in touch with a systems architect they know has experience in this area to help simplify the programming for these regulations.
- **Example from a “design, build, test, accept, release” model** These steps describe the common iterative cycle process that continues throughout the project. At any time during this process opportunities for improvement in story decomposition can emerge. For example, let’s say during the build step two team members are practicing paired programming. (This means two programmers sit together and take turns with one person programming while the other watches.) What if the observing programmer on the “manage account” story in figure 7.17 suddenly realizes that the stories for this feature can be broken down and built in a different way that results in a better, simpler design? Although perhaps invisible to the end user, the product has just been improved.

### Methods for Agile Story Decomposition

You probably know intuitively that there can be many methods of breaking down features and functions of a product, but you may not be aware of these less obvious methods. Here is a list of some common story breakdown methods.

- **Process-based breakdown** The “manage account” story in figure 7.17 shows a process-based breakdown. The process described as “Manage account” is shown to have three sub-processes associated with it (although there are more than the sub-processes shown in this simple example).
- **CRUD (Create, Read, Update, Delete)** This acronym stands for the list of things a programmer wants to allow a user to do with data. See the illustration in figure 7.17. We provide this example to help you remember the term CRUD, which may be used on the exam. You will not need to know more than this about this method.
- **Business rule-based breakdown** An example of a business rules breakdown might sound like this: “There are many ways for a patient to pay a bill. On the website we only accept the use of VISA, Mastercard, bank information for automatic withdrawals (if a payment plan is needed), and PayPal. For this example, the team will break down stories based on what it takes to build in this functionality.
- **User or platform-based breakdown** An example of the need for this type of breakdown is “We have to make this product easily usable on desktop PCs, laptops, tablets, and mobile devices.”
- **Acceptance test breakdown** An example of this is employing acceptance test-driven development (ATDD). With ATDD, acceptance tests are built before the story is built and then the story is built to pass the test.
- **MoSCoW analysis** This is a breakdown method of higher-level requirements at the release map and product (feature) backlog levels. MoSCoW stands for “Must have, Should have, Could have, and Would like to have,” and is a prioritization scheme for selecting features and functionality.

Figure 7.17 shows how “manage account” may be broken down using CRUD. Notice that “D” is not included since the clinic will not want a patient to be able to delete their own account.

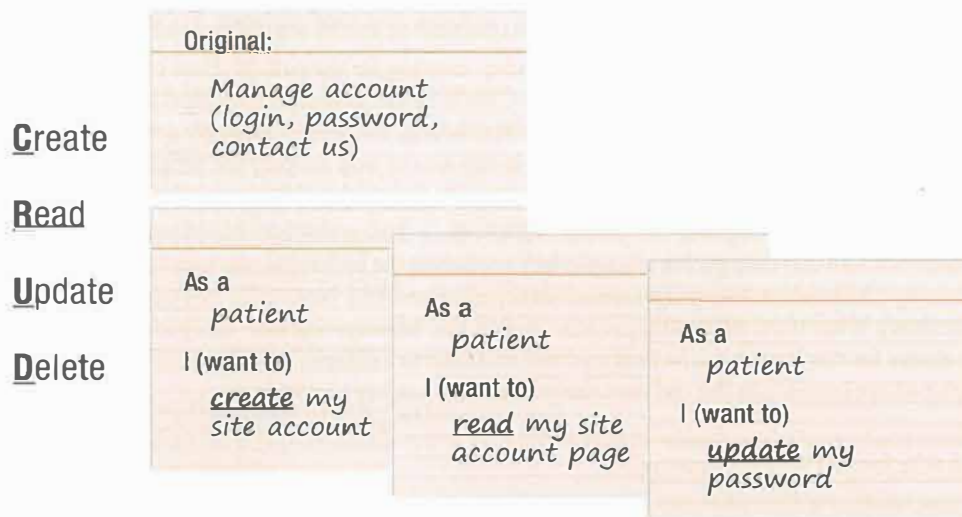


FIGURE 7.17 CRUD functional decomposition

## Validate and Control Scope

Many people are confused about what it means to validate scope. If you correctly understand scope control and validation from the perspective of the Process Groups model, you can get more questions right on the exam. According to the Process Groups model, validate and control scope are two distinct processes, but let's review the process of validating scope together with the process of controlling scope so you understand them holistically and are ready for related exam questions.

### TRICKS OF THE TRADE

First understand where in the project management process scope is validated. Many people think scope validation means confirming the validity and appropriateness of the scope definition during project planning. The Validate Scope process actually involves frequent planned meetings with the customer or sponsor to inspect and gain approval of deliverables during project control. That's a big difference, isn't it?

#### Process Groups Model

PG: Monitoring & Controlling  
Process: Validate Scope; Control Scope

#### ECO

Domain II  
Task 8 Plan and manage scope

#### PMBOK® Guide

Domain 2.6 Delivery  
Domain 2.7 Measurement

### Validate Scope

This is ideally the result of all project work—accepted deliverables! Validate Scope means taking already verified results to the customer. The customer will either accept deliverables or make change requests. The successful process culminates in the customer signing off on the results. This process is repeated throughout the project as interim deliverables are completed until the end of the project when the customer signs-off on (or validates) the final, delivered product—or on agile projects, the minimal viable product.

### Control Scope

The project manager and team control scope throughout the project—before, in concert with, and after the validate scope process. This, then, for the project manager is about monitoring progress, looking for ways to remove impediments to the team who are completing the scope in the allotted time and cost. This is how the project manager can help the team ensure that Control Quality will bring about expected, verified results, and the Validate Scope process can happen without difficulty. It involves measuring and assessing work performance data against the scope baseline and managing scope baseline changes.



What are the inputs to validate scope? Here's an exercise to look at the inputs to this process.

## 7.5 Exercise

In your Exercise Notebook, write the answer to this question: "What do I need before I can validate scope?"

### Answer

- **Verified deliverables** Validate Scope is intimately tied to Control Quality. Completed work must be checked before meeting with the customer. The deliverables are verified in Control Quality.
- **Scope baseline** This is needed (from the project management plan) for comparison. It's helpful to have the approved scope when meeting with the customer.
- **Scope management plan** This plan shows and plans for gaining formal acceptance of approved deliverables (described in the scope baseline).
- **Requirements management plan and requirements traceability matrix** The project manager exchanges information about the requirements and shows the customer how they have been validated. Comparing the requirements to results will help to determine if any action or change is needed.
- **Work performance data** This data from the Direct and Manage Project Work process helps the project manager assess how well product deliverables are meeting the requirements.
- **Other project documents** Quality reports and lessons learned should also be reviewed at the start of this process. Quality reports can include information about open or closed issues. Lessons learned can be used to improve the process of validating project deliverables.

### TRICKS OF THE TRADE

Did you notice that we didn't just list what the project manager needs to do but described how each artifact will be used? Whenever you think about the inputs of a process, make sure you can describe them and explain where they come from and what they offer to completing the process. Similarly, make sure you understand how outputs flow logically from each process. For the exam, this deeper understanding will often give you more insight into situational questions, help you distinguish between relevant and extraneous data, and help you select the correct answers. As you study, this understanding will spare you the need to memorize lists of terms like those used to name inputs, outputs, and tools and techniques.



**Think About It.** Did you happen to notice that there are no executing processes in the Process Groups model for the project manager for scope? This will be true of schedule and cost processes too, and it is because the team is responsible for the executing processes of scope, schedule, and cost. They are building the product (and spending the time and money to do so).

Now try an exercise on the efforts and outputs (artifacts) of these processes.



## 7.6 Exercise

In your Exercise Notebook, list what you are doing to control and validate scope, and what you have when you're done with the same processes.

## Answer

### What to Do to Control and Validate Scope

- Help the team focus on approved scope; do not add extras
- Help the team build from the top of a prioritized list like a backlog
- Collaborate with the team on ensuring a common understanding of scope
- Remove impediments for the team
- Gather and analyze work performance data
- Work on continuous improvement of processes and product quality
- Compare the deliverables to the requirements to make sure they meet stakeholder needs.

### What We Have When We Are Done with Control and Validate Scope

- Change requests
- Accepted deliverables
- Updates to project artifacts
- Work performance information
- Possible updates to processes and procedures

## Methods for Control and Validate Scope

Traditional methods for controlling and validating scope in predictive environments are based on observation and analysis. They include:

**Inspection** Product inspection is a routine part of controlling and verifying scope internally before validating scope with the customer and is in fact part of Control Quality.

**Data Analysis** is used in Control Scope. It includes variance analysis, which is comparing the scope baseline to actual project results to determine if variances are within acceptable limits. Related to this over a longer period of time is trend analysis, which helps tell the project manager and team if project performance is improving or worsening.

**Decision Making** Based on the data the project manager and team observe and analyze, and inspection of the workings of the product, the project manager and team can make decisions, largely based on consensus. Decisions may include how to handle issues, work around or fix problems on the spot, and otherwise prepare the product increment in question to either go to the customer for validation or remain in development until all issues are resolved.

### Controlling and Validating Scope in an Agile Environment

On a change-driven project, controlling, verifying, and validating scope happens at the end of each iteration as part of the iteration review with the customer. Let's say the team, in collaboration with the customer, has settled on doing three, two-week iterations, plus a "hardening off" iteration (to make sure everything is ready for release) before each product release. By the time an MVP is ready for the first product release, the team and customer have participated in three or four iteration reviews (if the hardening off iteration is included, and it typically is) where the customer has seen the increasingly mature product release before it is delivered to the marketplace.

Agile  
Focus

**Agile Ceremonies (Meetings)** Throughout the project and during every iteration the team has daily standup meetings to report to each other what they have been working on and have completed, what they will continue working on toward completion, and whether there are any impediments to progress. At the end of each iteration the team meets with the customer to demo and discuss what they have built, hopefully getting acceptance of that iteration's work but also taking back any customer feedback with which to improve the product increment before delivery. The team follows up each iteration review with an iteration retrospective among themselves to further the goal of continuous improvement in all

their processes, and particularly in improving the product of the current project. Agile ceremonies are focused on controlling and producing project scope.

**Customer-valued Prioritization** The role of the product owner is about prioritizing the product backlog according to customer priorities. The product owner represents the end users of the product and must bring anyone into the conversation who is important to the continuous delivery of the features the end-users (or customers) most value. It is the team's job then to help the product owner maximize that value with advice on technical requirements and risks. These requirements and risks are added to the backlog and integrated with the customer requirements for the product.

**Incremental Product Delivery** Through frequent product releases the team delivers minimal marketable features (MMFs) to the customer until the agreed-upon project backlog is complete. The product backlog usually continues to exist across several or many projects, and for some kinds of products, like software, smaller upgrades can be delivered on a regular basis as part of product maintenance.

## Artifacts of Control and Validate Scope

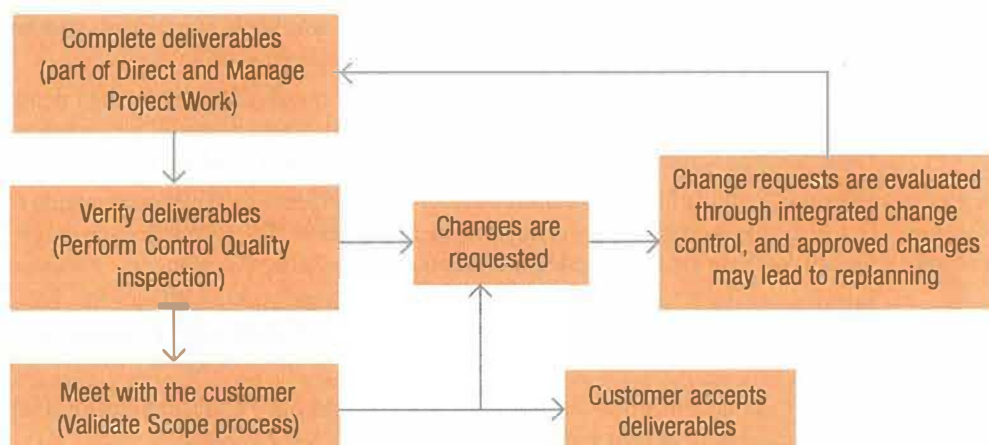
The artifacts of the Control and Validate Scope processes include:

- Project management plan and document updates
- Work performance information (analyzed work performance data)
- Change requests

For Validate Scope the resulting artifacts also include accepted deliverables. This means approval and formal sign-off by the customer, and may happen the first time the team shows a deliverable (product increment) to the customer, or only after changes have been made as a result of customer feedback.

There are a few more aspects to remember about Validate Scope:

- It can be done at any point in the project (within a phase) to get formal acceptance of interim deliverables that require approval (as part of monitoring and controlling). On agile it is done at the end of each iteration.
- It is done at the end of each project phase to get formal acceptance of interim deliverables.
- It is done at the end of a planned product release in agile.
- The difference between the Validate Scope and the Close Project or Phase processes can be a little tricky.
  - ✓ The Validate Scope process results in formal acceptance by the customer of deliverables.
  - ✓ The Close Project or Phase process is an integration process, to get final acceptance from the customer for the project or phase as a whole including not just product scope but project and phase closing activities, like indexing and archiving records, for example.
- We have already mentioned that Validate Scope and Control Quality are related. The high-level diagram in figure 7.18 should help you visualize this.



**FIGURE 7.18** Relationship of Validate Scope to Control Quality

Although Control Quality is generally done first to verify that the deliverable meets requirements before it is shown to the customer, the two processes are similar. Both involve checking for the correctness of work according to requirements. The focus is on who is inspecting and approving.

- In Control Quality, quality control checks to see if the requirements specified for the deliverables are met.
- In Validate Scope, the customer checks and hopefully accepts the deliverables.

As you take the exam, assume that the project manager is controlling scope to make sure the scope is being completed according to the project management plan. There should be a clear definition of and elaboration on scope in the scope baseline. Assume proper project management is being done on the project unless the question states otherwise.

The project manager then has to measure the completed work against the scope baseline, perform data analysis, including analyzing any variances, and determine whether the variances are significant enough to warrant changes. If necessary, they would submit a change request through the Perform Integrated Change Control process to assess the impact the change would have on all aspects of the project. New work performance information may result, along with updates to the project management plan and project documents.

Remember that the Control Scope process is proactive. It includes thinking about where changes to scope may be coming from on the project and what can be done to prevent or remove the need for any more changes from that source. Properly using project management tools, techniques, and practices will save you from unnecessary problems throughout the life of a project.

## Scope: Putting It All Together

### 7.7 Exercise

In our library case study, the project manager who is overseeing the creation of the new community library has gathered requirements, but the stakeholders differ on what is needed (or not needed) and on what they would like to see in the new library. How might the project manager work to resolve these competing requirements?

Stakeholder(s)	Requirement(s)
1. City Council member	A small coffee shop included in the library so people can meet and talk and have a drink/snack.
2. Librarian	No food or drink should be allowed or encouraged in the library. Spills damage books and technology, and costs more for cleaning every night.
3. Library staff	A small kitchen/break room where staff could refrigerate and heat lunches to decrease the need to go out for lunch which is costly and takes time. Also beneficial to patrons; sometimes a parent wants to heat their baby's food for story time without leaving the library.
4. Librarian	Against a kitchen because of additional mess and cleaning costs. She thinks the staff can simply bring cold sandwiches if they want to bring their lunch. Many employees only work part time so don't take a lunch break.
5. Mayor	Wants everyone who uses the library to login to the system and provide their demographic info (name, address, phone, email). They think this data will be useful for increasing voter registration and participation.
6. Citizens group advocating for online privacy	Does not want the library to collect demographic information. They want people to be able to use the library without providing private information. They think the mayor is trying to build a database for campaigning.
7. IT Security Consultant	For security, the least amount of information needed should be collected. The system should also require users to view a short security video before they are allowed to use the system.
8. Librarian	Some patron information is needed. If a patron checks out a book, and does not return it, reminders need to be sent and possibly late fees need to be charged.



## Answer

Here are some sample answers. You may have come up with some other solutions. Just make sure you understand our solutions, and that your solutions will help the situation.

Stakeholder(s)	Requirement(s)	Ways to resolve
1. City Council member	Small coffee shop be included in the library so people can meet and talk and have a drink/snack.	– Interview architect and construction teams; get high-level cost estimate of a coffee shop.
2. Librarian	No food or drink should be allowed or encouraged in the library. Spills damage books and technology, and cost more for cleaning every night.	– Send survey to include opinions of citizens (potential library patrons). – Assist the librarian to research estimates for additional cleaning costs/damage costs expected. – Get bids from coffee shop managers who would run the operations. – Have a brainstorming session about a coffee shop; generate ideas for decreasing or eliminating library materials damage.
3. Library staff	A small kitchen/break room; staff could refrigerate and heat lunches to decrease the need to go out for lunch which is costly and takes time. Beneficial to patrons; sometimes a parent wants to heat up baby's food for story time without leaving the library.	– Interview architect and construction teams; get a high-level estimate of the cost of a breakroom with kitchen. – Have a workshop with the current staff and librarian to talk about the pros and cons of this idea. Would the staff be willing to clean the kitchen?
4. Librarian	Against a kitchen because of the additional mess and cleaning costs. She thinks the staff can simply bring cold sandwiches if they want to bring their lunch. Many employees only work part time so don't even take a lunch break.	– Analyze current and future staffing needs: part time vs. full time.
5. Mayor	Would like everyone who uses the library to login and provide demographic info (name, address, phone, email). The mayor thinks this data will help increase voter registration and participation.	– Library software system can be built using an agile approach. – A backlog of requests will be compiled and prioritized at workshops with the stakeholders.
6. Citizens group advocating for online privacy	Does not want the library to collect citizen information. They want people to be able to use the library without providing private information. They think the mayor is trying to build a database for campaigning.	– A product owner could help the various stakeholders reach consensus on the next requirement meet.
7. IT Security Consultant	For security, the least amount of information needed should be collected. The system should also require users to view a short security video before they are allowed to use the system.	– MVP could be basic searches for materials and the security video. Future releases could consider some collection of data as the need arises.
8. Librarian	Some patron information is needed. If a patron checks out a book, and does not return it, reminders need to be sent and possibly late fees need to be charged.	