Managing Project Requirements

Course Materials

The causes of the majority of failed IT projects can be traced to poor definition and understanding of requirements. Changing requirements causes more delays than any other cause. If you want to improve project performance you must improve the way you define and manage requirements. Requirements must be a clear description in a common language. They are the bridge between the customer who will describe, pay for, and use the solution and the technical community who will specify and provide the solution. Requirements must be unique, normalized, linked, complete, consistent, bounded, modifiable, configurable and granular. Participants will cultivate the skills needed to elicit full business information from the right stakeholders and, learn structured techniques to identify and present business requirements clearly and effectively.

Table of Contents

Course Agenda

Day 1		Day2	
8:30 – 9:00	Personal Introductions	8:30 - 10:00	Defining Project Requirements
9:00 - 10:30	Intro to Requirements Mgmt	10:00 - 10:15	BREAK
10:30 - 10:45	BREAK	10:15 - 11:45	Non-Func and Transitional Reqmts
10:45 - 11:00	Intro to Requirements Mgmt	11:45 - 12:45	LUNCH
11:00 - 12:00	Determining Busn and Cust Needs	12:45 - 1:15	Non-Func and Transitional Reqmts
12:00 - 1:00	LUNCH	1:15 - 2:15	Validating Project Requirements
1:00 - 2:00	Determining Busn and Cust Needs	2:15 - 2:30	BREAK
2:00 - 2:15	BREAK	2:30 - 3:30	Validating Project Requirements
2:15 - 4:00	Defining Project Requirements	3:30 - 4:00	Exam and Evaluation

LESSON 1: INTRODUCTION TO REQUIREMENTS MANAGEMENT

Topic 1: Requirements Definition

Topic 2: The Project Environment

Topic 3: Understanding the Project Charter

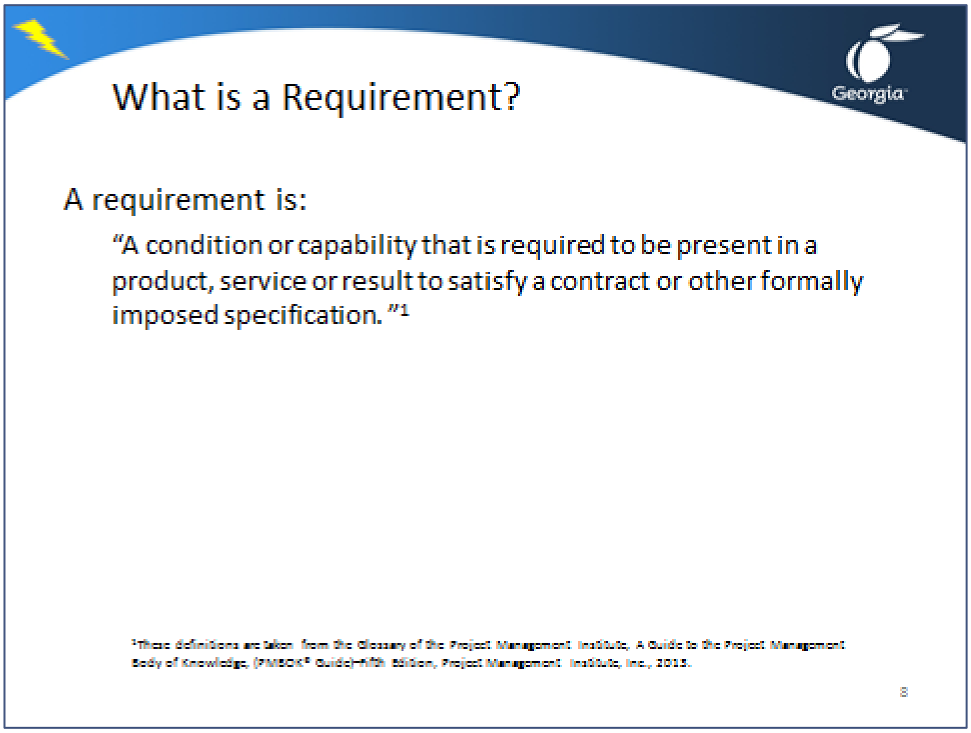
Student Learning Objectives

After completing this lesson you should be able to

- Identify what constitutes a requirement
- Explain the project environment for Requirements Management
- Describe the purpose and components of the Project Charter

Approximate Presentation time: 2 hour

Topic 1: Requirements Definition



What is a Requirement?

A requirement is:

"A condition or capability that is required to be present in a product, service or result to satisfy a contract or other formally imposed specification. "[1]

[1]These definitions are taken from the Glossary of the Project Management Institute, A Guide to the Project Management Body of Knowledge, (PMBOK® Guide)–Fifth Edition, Project Management Institute, Inc., 2013.

8

What is a requirement?

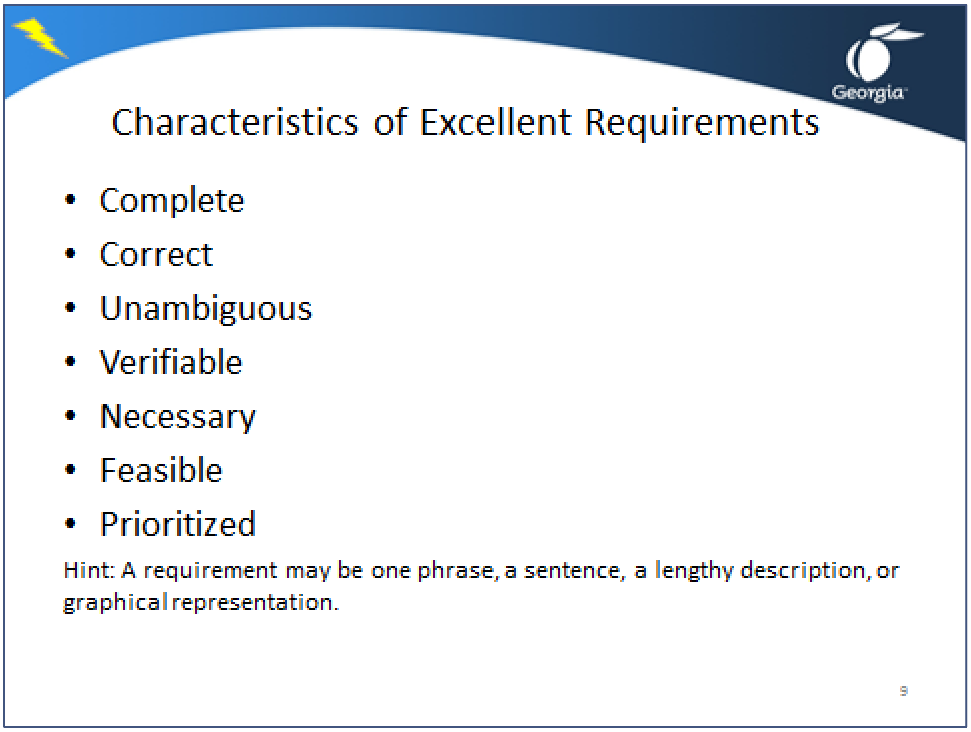
"A condition or capability that is required to be present in a product, service or result to satisfy a contract or other formally imposed specification.[1]"

A good requirement has the following characteristics:

- Complete
- Correct
- Unambiguous
- Verifiable
- Necessary
- Feasible
- Prioritized

[1] These definitions are taken from the Glossary of the Project Management Institute, A Guide to the Project Management Body of Knowledge, (*PMBOK® Guide*)–Fifth Edition, Project Management Institute, Inc., 2013.

Topic 1: Requirements Definition (continued)



Characteristics of Excellent Requirements

- Complete
- Correct
- Unambiguous
- Verifiable
- Necessary
- Feasible
- Prioritized

Hint: A requirement may be one phrase, a sentence, a lengthy description, or graphical representation.

9

Excellent Requirements are:

Complete: Make sure the requirement describes completely the user task and information required to support the task. Focusing on the system functionality versus ***what*** needs to be accomplished, may lead to incomplete requirements.

Example: “We must be able to change an employee’s profile information.”

If we don’t specify the individual components of the employee profile, this requirement is not complete. To make this requirement complete:

> “We must be able to change **the employee last name, first name, middle initial, street address, city, state, zip code, marital status, and/or withholding parameters**.”

Correct: The requirements should be appropriate to meet the goals of the project and accurately describe the user’s expectations of the functionality.

Example: “Employees only change their name when their address or their marital status changes.”

Someone who was not familiar with the business area may have assumed this requirement. Clearly this requirement is incorrect and must be changed.

> “Employees may change their name in the payroll system by providing the appropriate legal proof of the change. The change may come with a change in marital status, address, withholding parameters, or be made alone.”

Unambiguous: Requirements should be written so that all readers will arrive at a single, consistent interpretation. Ambiguous requirements can result in the wrong system being developed and may not be found during testing due to the incorrect interpretation of the requirement.

Example: “Employees are not allowed to work for more than 80 hours in one week.”

“Not allowed” is ambiguous. Are they physically removed from the work environment or are they not paid for any hours over 80?

> “Employee time worked: the time worked is recorded in hours, the smallest increment recorded is .25 of an hour, if an employee reports more than 80 hours in a seven day period a warning is provided to the supervisor and the payment is held for approval.”

Verifiable: Each requirement should be testable and verifiable.

Example: “The system should be easy to use.”

The requirement is impossible to test since every user will have a different opinion about what is easy and what is not.

> “A novice user must be able to add a new employee to the payroll system within 10 minutes.”

> “An experienced user must be able to change an employee’s withholding parameters within 3 minutes.”

Necessary: Requirements must be necessary and clearly support on of the original project goals or objectives.

Example: “We should be able to enter the employee eye color.”

This is a great example of a time when the Project Manager needs to ask the “WHY” question. Is this requirement necessary?

Feasible: The Project Manager should be sure that all requirements are technologically possible for a reasonable cost.

Example: “The system should automatically be updated when the government changes the withholding parameter criteria.”

Although this requirement may be technologically feasible, it would **involve a complex interface with an outside organization and may be very expensive**. Is it a critical requirement?

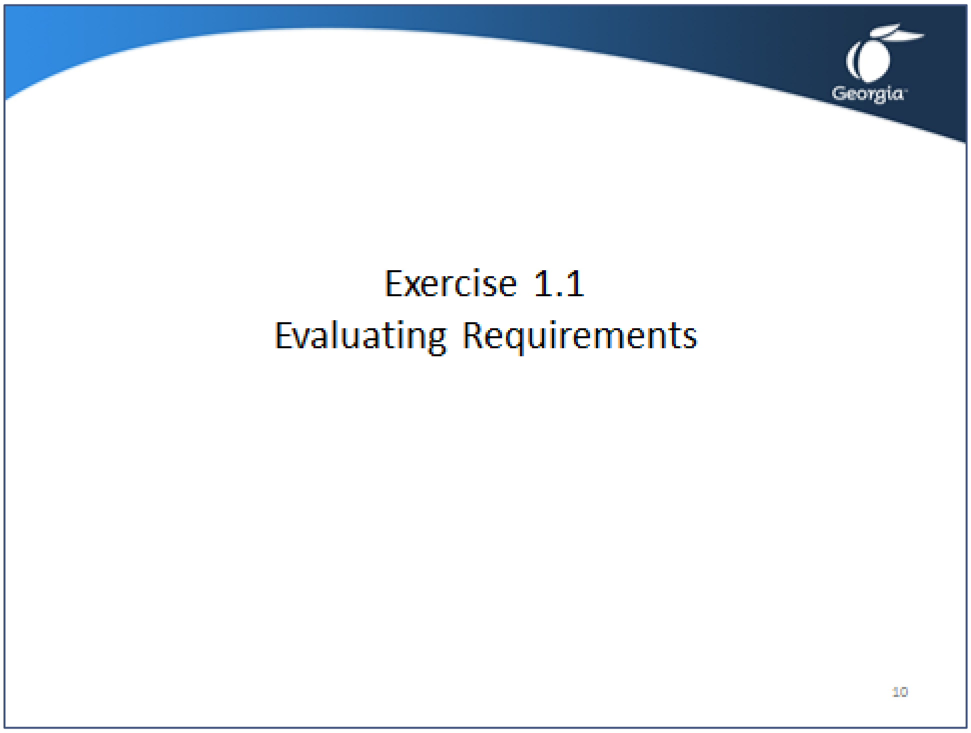
Prioritized: Each requirement should be prioritized. One approach is to have the Project Manager prioritize the project objectives and track the requirements related to each project objective.

Some organizations require one of the following phrases in each requirement:

- Must Have
- Should Have
- Could Have

Hint: Some approaches use the word “shall” instead of “should”. Be sure to define what the words “shall” and/or “should” mean in your organization.

Exercise 1.1 Evaluating Requirements



Exercise 1.1
Evaluating Requirements

10

Review each requirement below and describe why each requirement is not excellent.

Requirement
“A customer is any company that has done business with our organization in the past or potentially will in the future.”

Why is this requirement not Excellent?

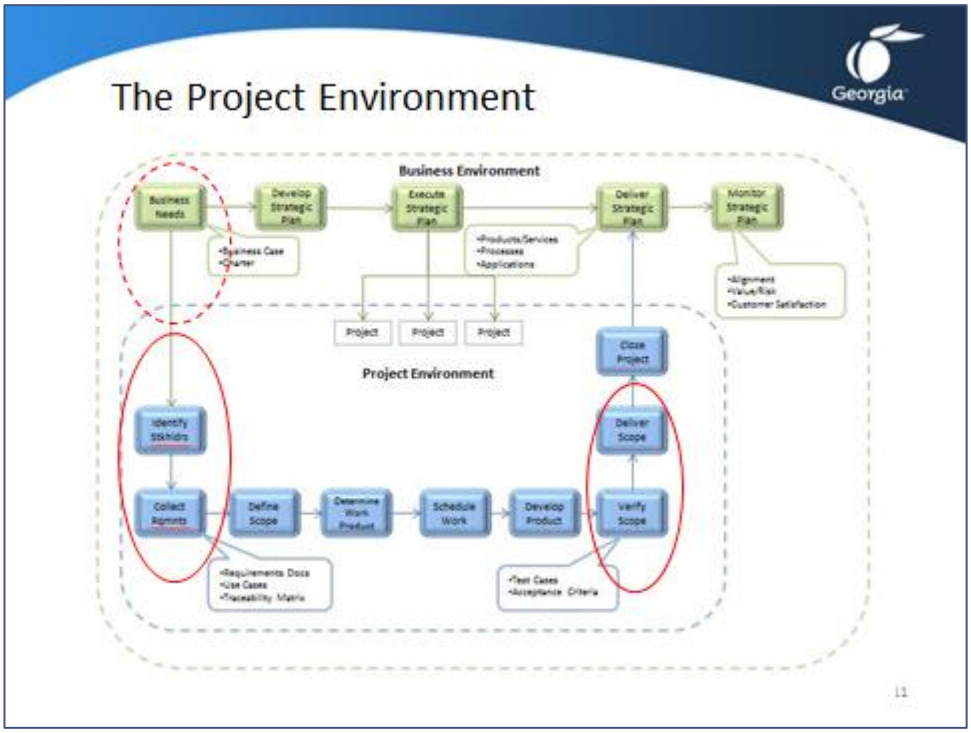
Requirement
“”For each of our customers, we need to know the names of each employee with which our organization has contact.”

Why is this requirement not Excellent?

Requirement
“Telemarketers should receive the next phone call on their screen as quickly as possible after ending the previous call.”

Why is this requirement not Excellent?

Topic 2: The Project Environment



The Project Environment

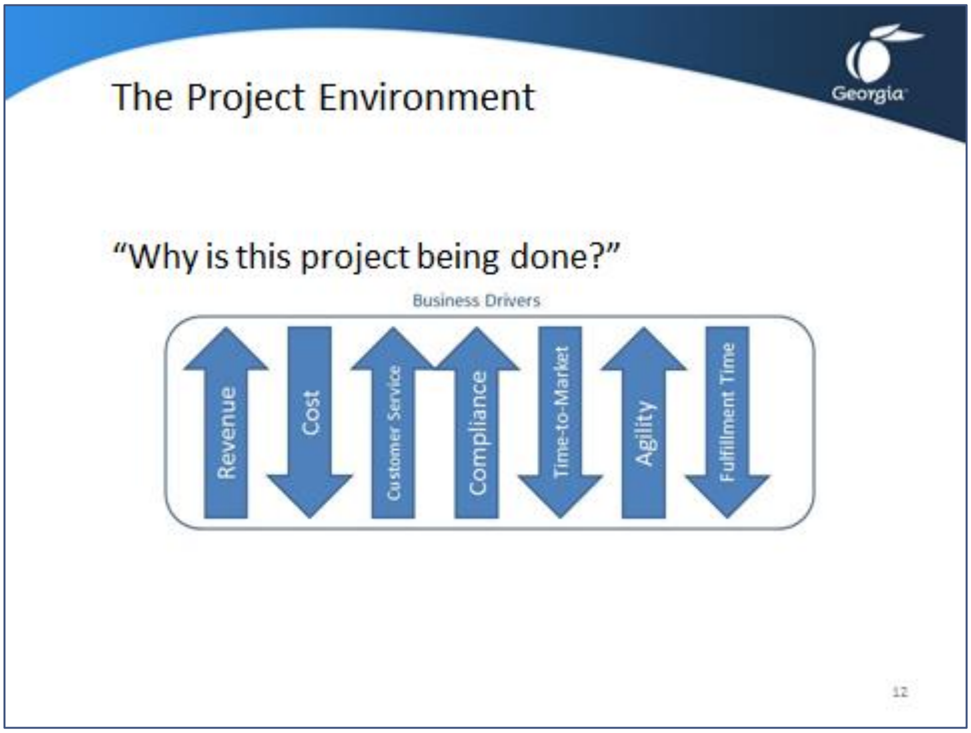
Projects are performed to solve a business problem or address an opportunity. Either an existing system (usually a combination of technical and business procedures) must be changed or replaced, or a new system created.

The definition of the problem, its “value” to the organization and the way it will be resolved, represent the information needed to establish project objectives.

It is very important for the project manager to understand the organizational environment he/she is operating in. Here are several things you need to be aware of:

- Which organizational units will be involved with the project?
- Which individuals within those units are assigned to the project?
- How will the various organizational units interact with each other?
- How will the SMEs interact with the technical team?
- What do the SMEs expect form the Project Manager?
- How will the technical team develop the systems?
- What will the technical team expect from the project manager?

Topic 2: The Project Environment (continued)



The Project Environment

"Why is this project being done?"

Business Drivers

Revenue

Cost

Customer Service

Compliance

Time-to-Market

Agility

Fulfillment Time

12

In addition to understanding the personnel and politics of the project, the Project Manager must know the importance of the project to the organization.

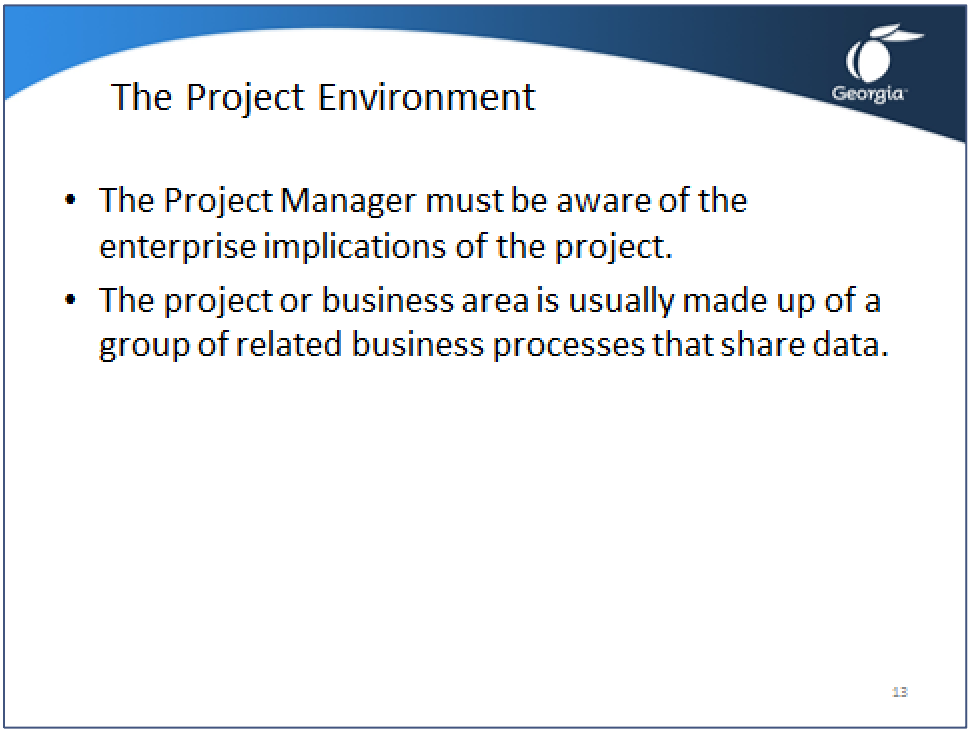
The most important question that the Project Manager must be able to answer is:

> "Why is this project being done?"

The answer to this question drives the decisions throughout the life of the project. The Project Manager must plan the work of the project around all of the stakeholders, taking into consideration various business drivers such as:

1. Increasing Revenues
2. Decreasing Costs
3. Increasing Customer Service
4. Increasing Compliance
5. Decreasing Time-to-Market
6. Increasing Agility
7. Decreasing Fulfillment time

Topic 2: The Project Environment (continued)



The Project Environment

- The Project Manager must be aware of the enterprise implications of the project.
- The project or business area is usually made up of a group of related business processes that share data.

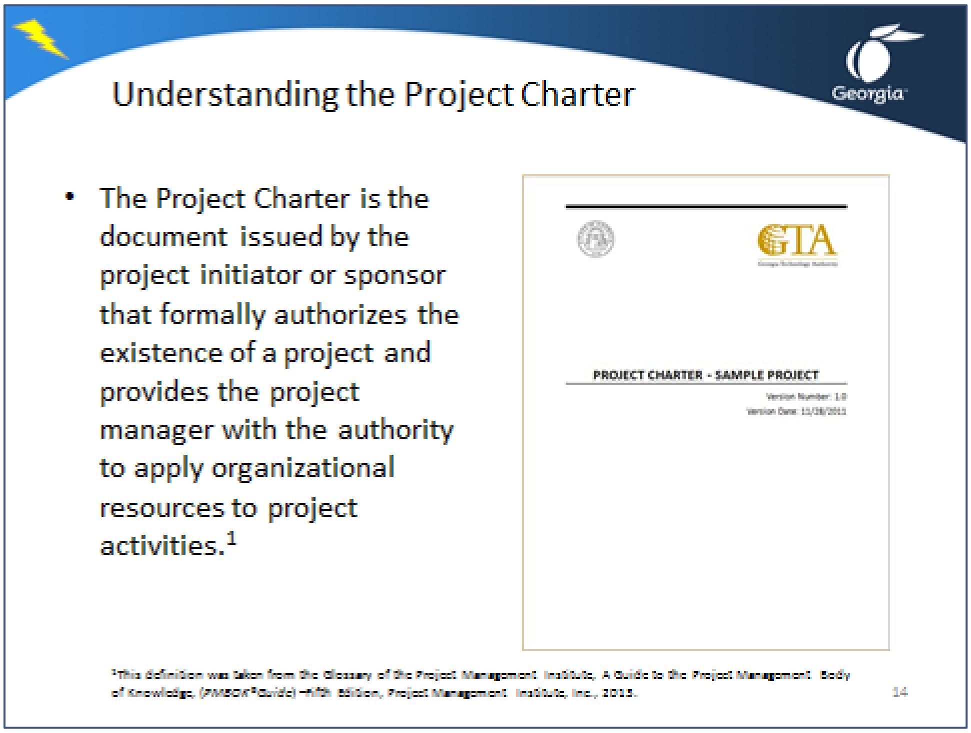
13

The Project Manager must be aware of the enterprise implications of the project. The project or business area is usually made up of a group of related business processes that share data. It may coincide with an organizational unit (a division, or department) or it may cross multiple organizational units.

For example: Project: Create an Enterprise Wide Customer Information System

Organizational Unit	Data Used	Business Process
Sales	Customer profile and product needs	Identify a new customer
Order Processing	Customer profile and products ordered	Accept customer order
Accounts Receivable	Customer profile, order information, customer credit	Bill customer
Shipping	Customer address, order information	Ship order

Topic 3: Understanding the Project Charter



Understanding the Project Charter

- The Project Charter is the document issued by the project initiator or sponsor that formally authorizes the existence of a project and provides the project manager with the authority to apply organizational resources to project activities.[1]

PROJECT CHARTER - SAMPLE PROJECT

Version Number: 1.0
Version Date: 11/28/2011

[1]This definition was taken from the Glossary of the Project Management Institute, A Guide to the Project Management Body of Knowledge, (PMBOK®Guide) –Fifth Edition, Project Management Institute, Inc., 2013.

14

The Project Charter is the document issued by the project initiator or sponsor that formally authorizes the existence of a project and provides the project manager with the authority to apply organizational resources to project activities.[1] It documents the business needs, assumptions, constraints, the understanding of the customer's needs and high-level requirements, and the new product, service, or result that it is intended to satisfy. The project charter is used as an input to Collecting Requirements to provide high-level requirements of the product, service, or result of the project so that the detailed requirements can be developed. Components of the Project Charter include:

- Project purpose or justification
- Measurable project objectives and related success criteria
- High-level requirements
- Assumptions and constraints
- High-level project description and boundaries
- High-level risks
- Summary milestone schedule
- Summary budget
- Stakeholder list
- Project approval requirements (i.e., what constitutes project success, who decides the project is successful, and who signs off on the project)
- Assigned project manager, responsibility, and authority level
- Name and authority of the sponsor or other person(s) authorizing the project charter.[2]

[1]This definition was taken from the Glossary of the Project Management Institute, A Guide to the Project Management Body of Knowledge, (*PMBOK®Guide*) –Fifth Edition, Project Management Institute, Inc., 2013.
[2] *PMBOK® Guide*, Page 71-72.

Project Charter Components

Executive Summary

The executive summary should be a high-level summary of what issues or problems the project was created to correct. Typically, the executive summary also provides the background information and general statements regarding the project's purpose or justification which will be covered in more detail in the appropriate section(s) of the charter.

Project Purpose/Justification

This section describes the purpose and justification of the project in the form of business case and objectives. The business case should provide the reasoning behind the need for this project as it relates to a function of the business.

Business Need/Case

Discuss the logic for the Business Need/Case (market demand, organizational need, customer request, technological advance, legal requirement, ecological impacts, social need, etc.). This section should also include the intended effects of the business case (i.e. cost savings, process improvement, new product development, etc.).

Business Objectives

This section should list the Business Objectives for the project which should support the organizational strategic plan.

Project Description

This section provides a high-level description of the project. This description should not contain too much detail but should provide general information about what the project is, how it will be done, and what it is intended to accomplish. As the project moves forward the details will be developed, but for the project charter, high-level information is what should be provided.

Project Objectives and Success Criteria

Objectives should be SMART: Specific, Measurable, Attainable, Realistic, and Time-bound. The project manager must be able to track these objectives in order to determine if the project is on the path to success. Vague, confusing, and unrealistic objectives make it difficult to measure progress and success.

Requirements

The project team should develop a list of all high-level project requirements. These requirements are clear guidelines within which the project must conform and may be a result of input from the project sponsor, customer, stakeholders, or the project team.

Constraints

Constraints are restrictions or limitations that the project manager must deal with pertaining to people, money, time, or equipment. It is the project manager's role to balance these constraints with available resources in order to ensure project success.

Assumptions

The project team must identify the assumptions they will be working under as the project goes forward. These assumptions are what the project manager/team expects to have or be made available without anyone specifically stating so.

Preliminary Scope Statement

The preliminary scope statement is a general paragraph which highlights what the project will include along with any high-level resource or requirement descriptions, and what will constitute completion of the project. This preliminary scope statement is exactly that: preliminary. All of this information will be expanded upon in greater detail as the project moves forward and undergoes progressive elaboration.

Risks

All projects have some form of risk attached. This section should provide a list of high-level risks that the project team has determined apply to this project.

Project Deliverables

This section should list all of the deliverables that the customer, project sponsor, or stakeholders require upon the successful completion of the project. Every effort must be made to ensure this list includes all deliverables and project sponsor approval must be required for adding additional deliverables in order to avoid scope creep.

Summary Milestone Schedule

This section provides an estimated schedule of all high-level project milestones. It is understood that this is an estimate and will surely change as the project moves forward and the tasks and milestones and their associated requirements are more clearly defined.

Summary Budget

The summary budget should contain general cost components and their planned costs. As the project moves forward these costs may change as all tasks and requirements become clearer. Any changes must be communicated by the project manager.

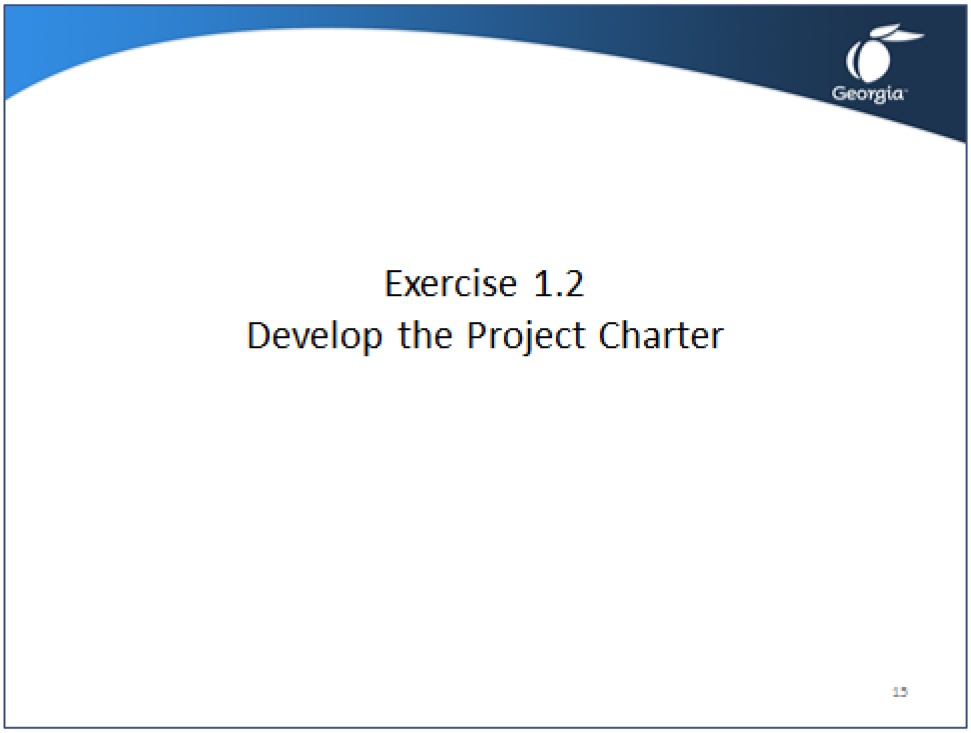
Project Approval Requirements

The organization must understand when the project has reached a successful completion. These criteria must be clear and should be accepted by whoever will sign-off on the project's closeout. Once signed-off by the authorized person, the project is deemed approved and is successful as long as it has met all of the agreed upon requirements.

Project Manager

This section explicitly states who is assigned as the PM, their responsibility, and authority level. Depending on the organization and scope of the project, the project manager may have varying levels of responsibility and authority for personnel, project expenditures, and scheduling.

Exercise 1.2 Develop the Project Charter



Exercise 1.2
Develop the Project Charter

15

Develop the Project Charter for the Case Study project using the components listed below.

Project Charter Components

Project Purpose or Justification

Project Objectives

Project description

High-level Requirements

High-level risks

Project manager, authority level

Lesson 1 Summary: Learning Objectives Recap

- **Identify what constitutes a requirement**
  A requirement is: "A condition or capability that is required to be present in a product, service or result to satisfy a contract or other formally imposed specification."

  A good requirement has the following characteristics:
  - Complete
  - Correct
  - Unambiguous
  - Verifiable
  - Necessary
  - Feasible
  - Prioritized
- **Explain the project environment for Requirements Management**
  Projects are performed to solve business problems. The project manager must understand the following areas:
  - The organizational environment: what organizations are involved; who will be assigned; how will the organizations interact with each other.
  - The business drivers for the project: why is the project being done?
  - The enterprise implications of the project; is the project related to an organizational unit or is it cross-functional.
- **Describe the purpose and components of the Project Charter**
  The Project Charter is the document issued by the project initiator or sponsor that formally authorizes the existence of a project and provides the project manager with the authority to apply organizational resources to project activities.

  It contains the following components:
  - business needs,
  - assumptions,
  - constraints,
  - the understanding of the customer's needs and high-level requirements, and
  - the new product, service, or result that it is intended to satisfy.

Notes

Lesson 2: Determining Business and Customer Needs

Topic 1: Understanding Customer Needs

Topic 2: Identifying Stakeholders

Topic 3: Develop the Requirements Management Plan

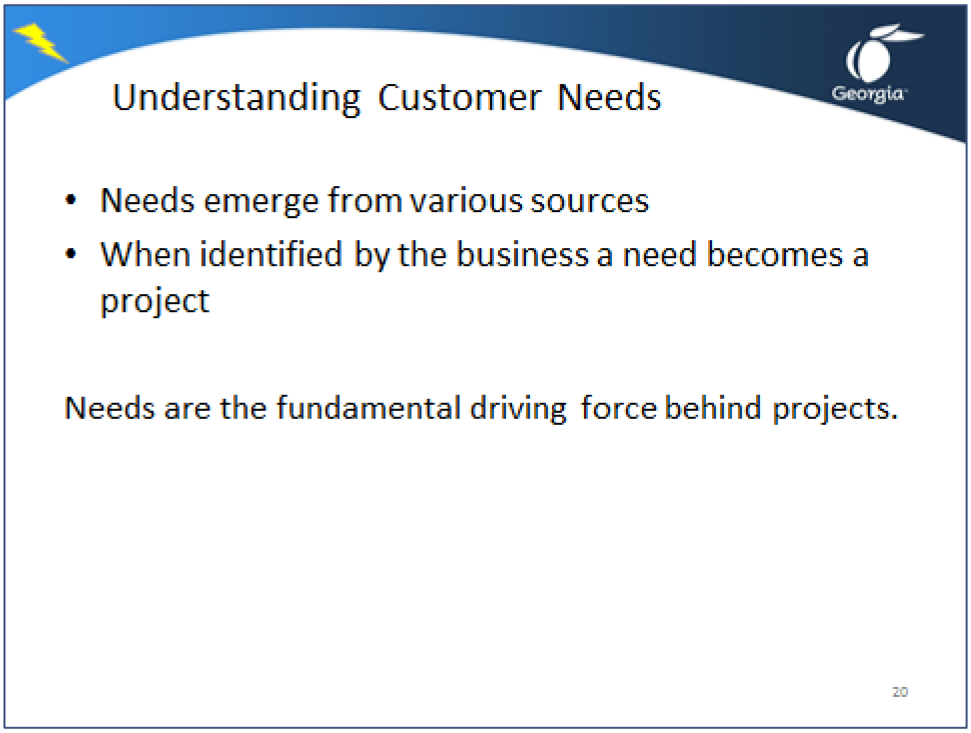
Student Learning Objectives

After completing this lesson you should be able to

- Describe how customer needs are identified and developed into requirements
- Explain the role of the stakeholder in determining requirements
- Discuss the components of the Requirements Management Plan

Approximate Presentation time: 2 hours

Topic 1: Understanding Customer Needs



Understanding Customer Needs

- Needs emerge from various sources
- When identified by the business a need becomes a project

Needs are the fundamental driving force behind projects.

20

Projects arise in order to meet human needs. A need emerges and is recognized, and then management determines whether the need is worth fulfilling. If it is, a project is organized to satisfy the need. Thus, needs are the fundamental driving force behind projects. The emergence of a need sets off the whole project process.

If at the outset we do not fully understand a need and its implications, if we incorrectly articulate it, or if we mistakenly address the wrong need, we have gotten off to a bad start and can be certain that our project will be trouble filled.

Needs evolve from something very vague to something well-structured and clearly understood. The case study on the next page will illustrate how needs evolve.

Ralph’s Drugstore

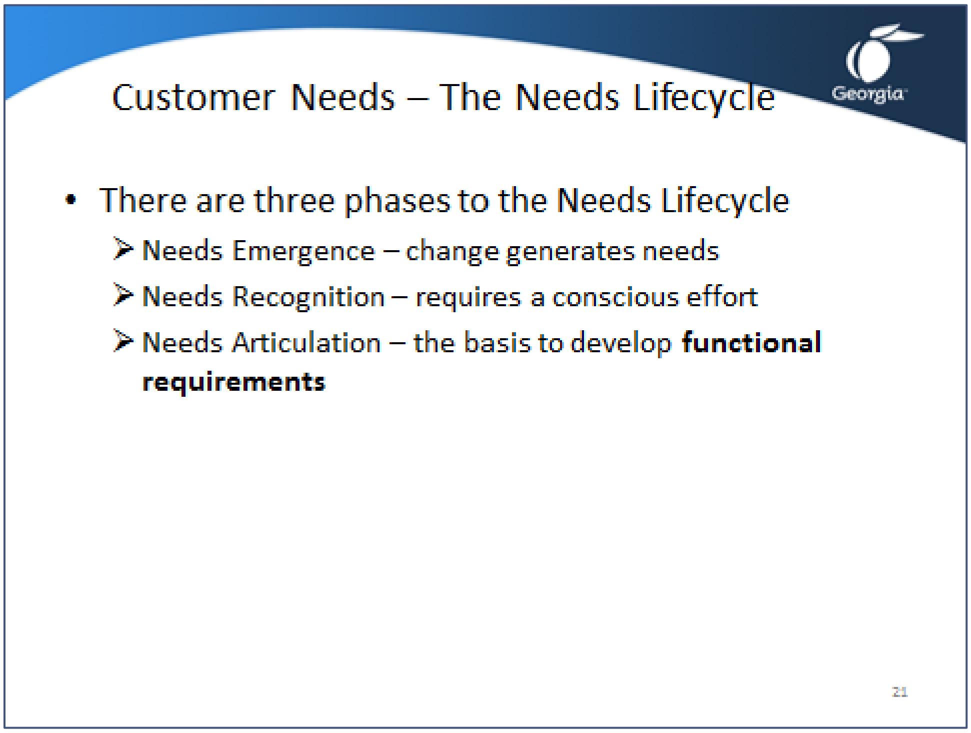
Ralph’s Drugstore is located in a small Midwestern town. While visiting Minneapolis on vacation, Ralph Amdahl, the owner, was impressed by the volume of business carried out by the city’s discount drugstores. When he returned home, he converted his drugstore into a discount operation, a complicated process that took six months.

Business volume soon increased dramatically. People would come from miles away to take advantage of Ralph’s discount prices. The store aisles were constantly jammed, and a long line snaked from the store’s single cash register. Ralph witnessed the crowds with mixed emotions. On the one hand, it was good to see that his discount policy was bringing in the crowds. On the other hand, customer dissatisfaction was growing. Complaints were primarily directed at three things: the crowded conditions in the store, stock-outs of special sale items, and long waits in the checkout line. Ralph was concerned that his success would backfire and that customer dissatisfaction with service would stymie growth.

Ralph expressed his concerns to Marie, his wife and business partner. One evening, the two of them sat down after dinner to discuss the future of the business. They determined that although their discount business was dramatically different from their previous operation, the basic way they conducted their business had not changed. For example, the physical layout of the store was no different than it had been before, and it was now apparent that this layout was inadequate to deal with the growth in customer traffic. Ralph and Marie decided that they needed more floor space, more shelf space, more cash registers, and more sales staff. They would have to either; remodel their current facilities, build new facilities, or rent different facilities. With paper and pencil, they roughly calculated their requirements: to meet anticipated customer traffic, they would need to double floor space and shelf space and add at least two cash registers. They concluded that to satisfy these requirements they would have to move to new facilities.

At this point, they met twice with a local architect to identify how best to configure a store to make it better suited to the new kind of business they were doing. Using the information gathered from these meetings, the architect designed three different store configurations. Ralph and Marie were excited by the second design, which required them to build rather than rent a new structure; and, after suggesting some minor modifications to the plan, they authorized the architect to proceed to make detailed drawings of the new facility. He completed the architectural plans within six weeks, and three months later ground-breaking for the new store was initiated.

Topic 1: Understanding Customer Needs-Needs Lifecycle



Customer Needs – The Needs Lifecycle

- There are three phases to the Needs Lifecycle
  - ➢ Needs Emergence – change generates needs
  - ➢ Needs Recognition – requires a conscious effort
  - ➢ Needs Articulation – the basis to develop **functional requirements**

The case of Ralph's Drugstore illustrates the evolution of needs from something vague to something tangible that serves as the basis of a project plan. There are three phases in the Needs Lifecycle.

1. Needs Emergence: Change is the generator of needs. Needs can arise from within or outside the organization.

2. Needs Recognition: If a need is not seen to exist, no action will be undertaken to satisfy them. Recognition of needs requires a conscious effort. Effective recognition also requires forecasting.

3. Needs Articulation: Needs must be clearly articulated. Articulation also serves as the basis to develop functional requirements. Follow these steps.
    a. See the need through the customer's eyes
    b. As a full set of questions about the need
        i. How do those who have the need define it?
        ii. Is the need real? Is this a true need or is it masking a more basic need?
        iii. Can we resolve the need? Can someone else resolve it? Is it resolvable at all?
        iv. Is the need important? Is it worth trying to satisfy?
        v. What are the implications of the need? If resolved, will it give rise to other needs? Will we satisfy other needs?
        vi. Who are the actors that are most directly touched by the need?
    c. Carry out any necessary research to understand the need better
    d. Formulate the need as clearly as you can
    e. Ask for customer feedback on your formulation and revise if needed

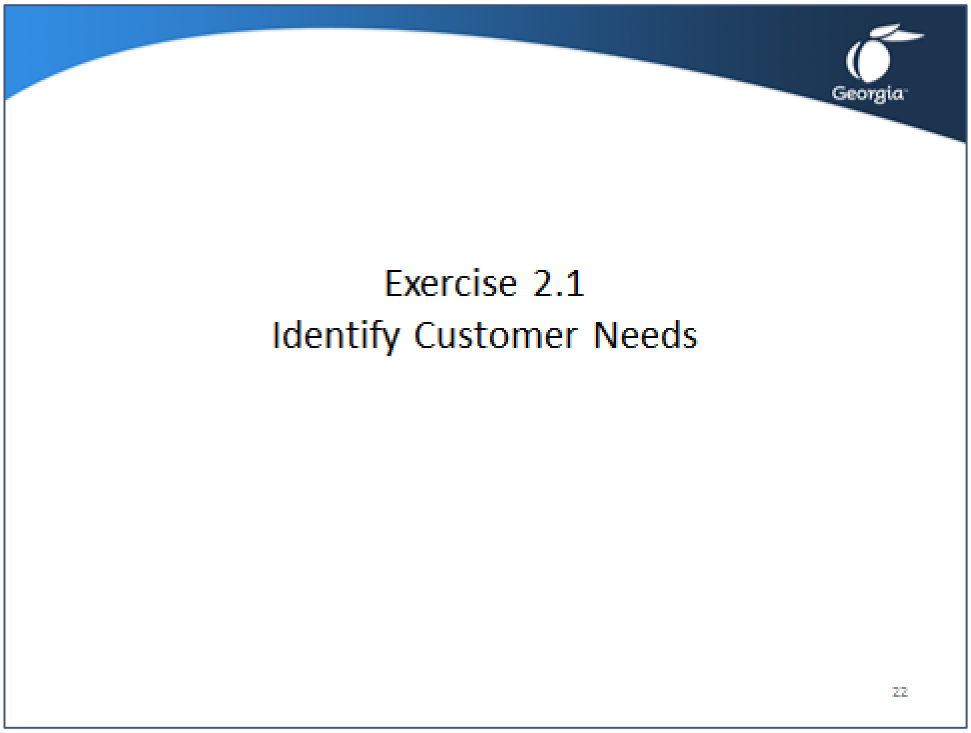
Looking at the case study on Ralph’s Drugstore, identify what part of the case study corresponds to the phases of the Needs Lifecycle.

Needs Emergence:

Needs Recognition:

Needs Articulation:

Exercise 2.1 Identify Customer Needs



Exercise 2.1
Identify Customer Needs

22

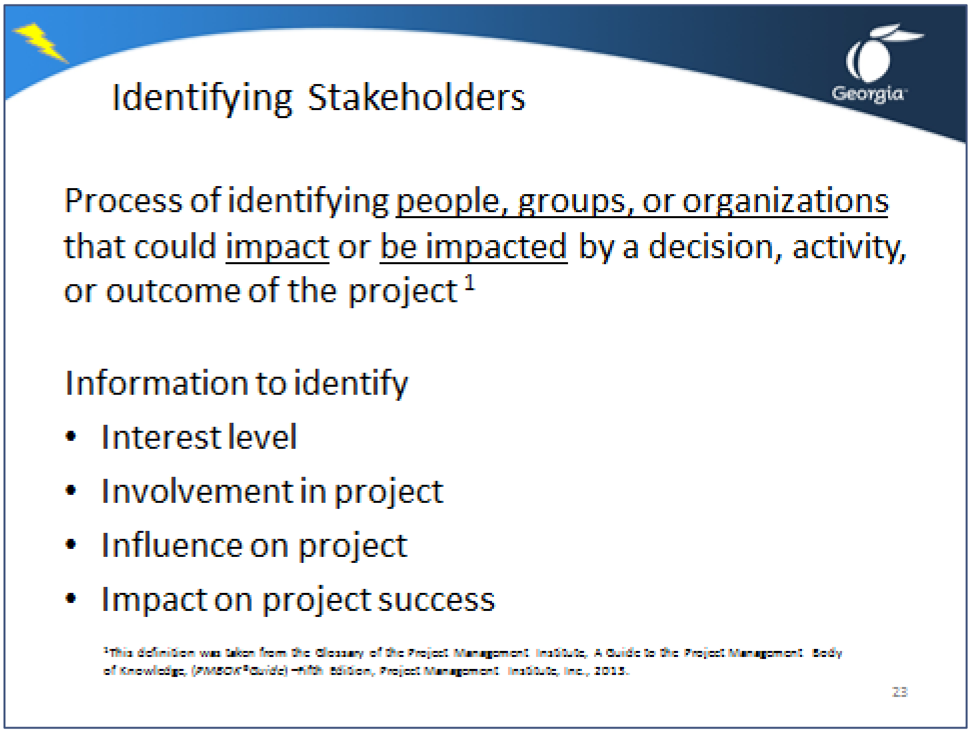
Using the information provided in the Case Study, determine the customer needs using the Needs Lifecycle.

Needs Emergence:

Needs Recognition:

Needs Articulation:

Topic 2: Identifying Stakeholders



Identifying Stakeholders

Process of identifying people, groups, or organizations that could impact or be impacted by a decision, activity, or outcome of the project [1]

Information to identify

- Interest level
- Involvement in project
- Influence on project
- Impact on project success

[1] This definition was taken from the Glossary of the Project Management Institute, A Guide to the Project Management Body of Knowledge, (*PMBOK®Guide*) –Fifth Edition, Project Management Institute, Inc., 2013.

23

Identifying Stakeholders is the process of identifying the people, groups, or organizations that could impact or be impacted by a decision, activity, or outcome of the project; and analyzing and documenting relevant information regarding their interests, involvement, interdependencies, influence, and potential impact on project success. [1]

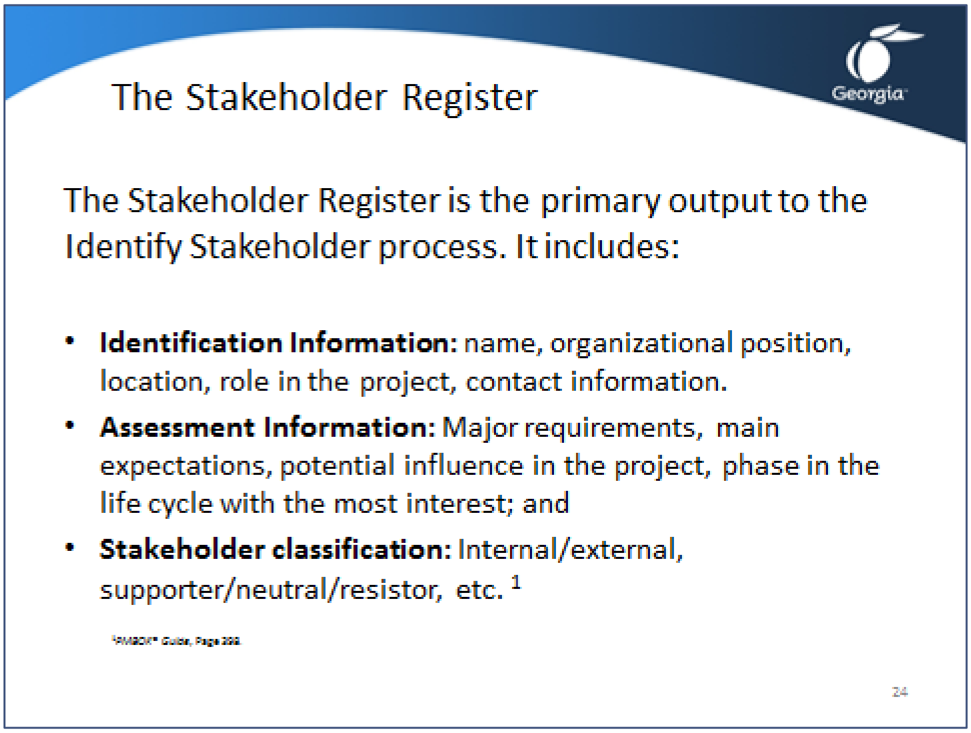
Stakeholders are comprised of customers, sponsors, the performing organization, and the public who are actively involved in the project, or whose interests may be positively or negatively affected by the execution or completion of the project. They may also exert influence over the project and its deliverables. Stakeholders may be at different levels of the organization and may possess different authority levels, or may be external to the performing organization of the project.

It is critical for project success to identify the stakeholders early in the project or phase and to analyze their levels of interest, their individual expectations, as well as their importance and influence. These stakeholders should be classified according to these criteria along with their level of involvement in the project. [2]

[1] This definition was taken from the Glossary of the Project Management Institute, A Guide to the Project Management Body of Knowledge, (*PMBOK®Guide*) –Fifth Edition, Project Management Institute, Inc., 2013.

[2] *PMBOK® Guide*, Page 394.

Topic 2: Identifying Stakeholders – Stakeholder Register



The Stakeholder Register

The Stakeholder Register is the primary output to the Identify Stakeholder process. It includes:

- **Identification Information:** name, organizational position, location, role in the project, contact information.
- **Assessment Information:** Major requirements, main expectations, potential influence in the project, phase in the life cycle with the most interest; and
- **Stakeholder classification:** Internal/external, supporter/neutral/resistor, etc. [1]

[1] PMBOK® Guide, Page 398

24

The Stakeholder Register is the primary output of the Identify Stakeholders process. This contains all details related to the identified stakeholders including, but not limited to;

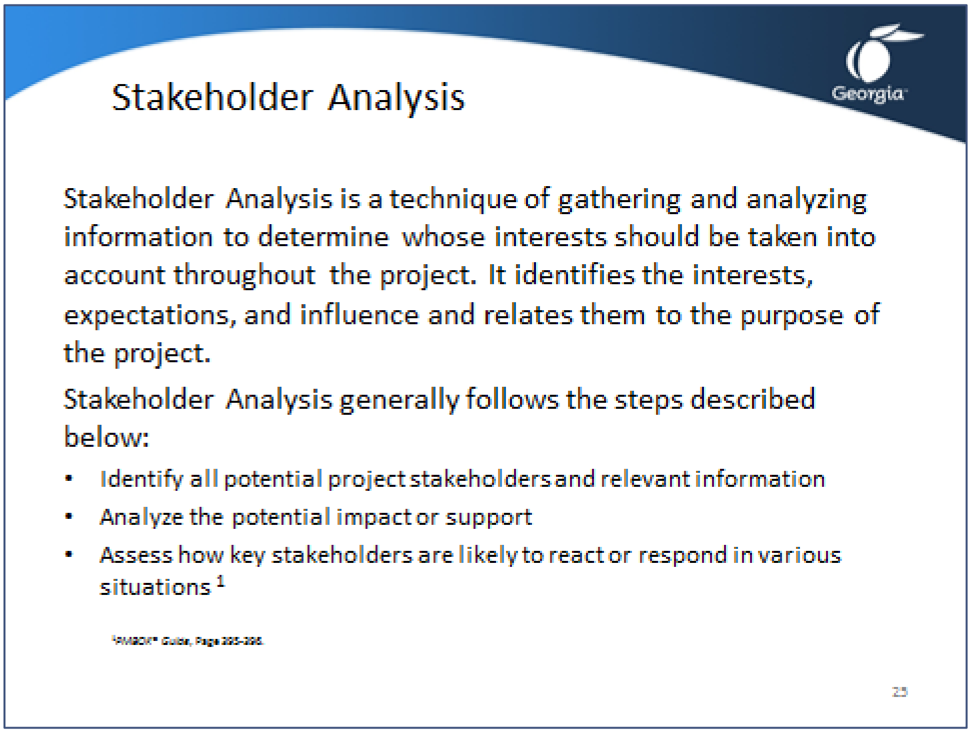
- **Identification Information:** name, organizational position, location, role in the project, contact information.
- **Assessment Information:** Major requirements, main expectations, potential influence in the project, phase in the life cycle with the most interest; and
- **Stakeholder classification:** Internal/external, supporter/neutral/resistor, etc.

The Stakeholder Register is used in the Collect Requirements process to identify stakeholder requirements and main expectations they may have in the project.

The Stakeholder Register should be consulted and updated on a regular basis, as stakeholders may change – or new ones identified – throughout the life cycle of the project. [1]

[1] *PMBOK® Guide*, Page 398

Topic 2: Identifying Stakeholders – Stakeholder Analysis



Stakeholder Analysis

Stakeholder Analysis is a technique of gathering and analyzing information to determine whose interests should be taken into account throughout the project. It identifies the interests, expectations, and influence and relates them to the purpose of the project.

Stakeholder Analysis generally follows the steps described below:

- Identify all potential project stakeholders and relevant information
- Analyze the potential impact or support
- Assess how key stakeholders are likely to react or respond in various situations [1]

[1] PMBOK® Guide, Page 395-396.

25

Stakeholder Analysis is a technique of gathering and analyzing information to determine whose interests should be taken into account throughout the project. It identifies the interests, expectations, and influence and relates them to the purpose of the project. It also helps to identify stakeholder relationships that can be leveraged to build coalitions and potential partnerships to enhance the project's chance of success, along with stakeholder relationships that need to be influenced differently at different stages of the project or phase.

Stakeholder Analysis generally follows the steps described below:

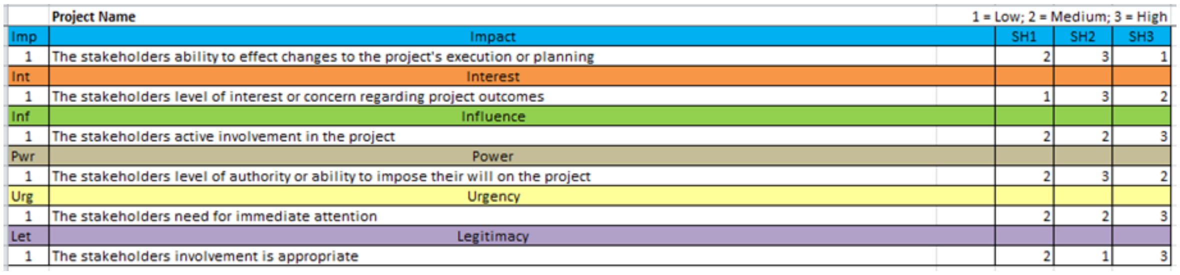
- Identify all potential project stakeholders and relevant information, such as their roles, departments, interests, expectations, and influence levels.
- Analyze the potential impact or support each stakeholder could generate, and classify them so as to define an approach strategy. It is also important to prioritize stakeholders to ensure effective communication and expectation management.
- Assess how key stakeholders are likely to react or respond in various situations, in order to plan how to influence them to enhance their support and mitigate potential negative impacts.

There are multiple classification models used for stakeholder analysis, such as:

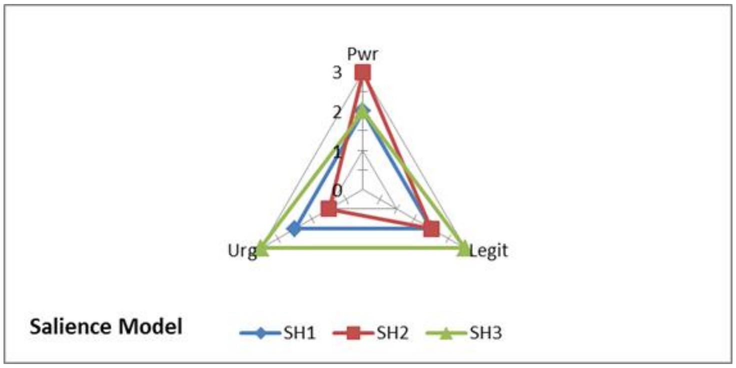
- Power/Interest: a grouping of stakeholders based on their level of authority ("power") and their level of concern ("interest") regarding project outcomes;
- Power/Influence: a grouping of stakeholders based on their level of authority ("power") and their active involvement ("influence") in the project;
- Influence/Impact: a grouping of stakeholders based on their active involvement ("influence") in the project and their ability to effect changes to the projects planning or execution ("impact"); and

- Salience model: describing classes of stakeholders based on their power (ability to impose their will), urgency (need for immediate attention), and legitimacy (their involvement is appropriate).

The tables and graphs below illustrate these models. [1]



	Project Name	1 = Low; 2 = Medium; 3 = High		
Imp	Impact	SH1	SH2	SH3
1	The stakeholders ability to effect changes to the project's execution or planning	2	3	1
Int	Interest			
1	The stakeholders level of interest or concern regarding project outcomes	1	3	2
Inf	Influence			
1	The stakeholders active involvement in the project	2	2	3
Pwr	Power			
1	The stakeholders level of authority or ability to impose their will on the project	2	3	2
Urg	Urgency			
1	The stakeholders need for immediate attention	2	2	3
Let	Legitimacy			
1	The stakeholders involvement is appropriate	2	1	3



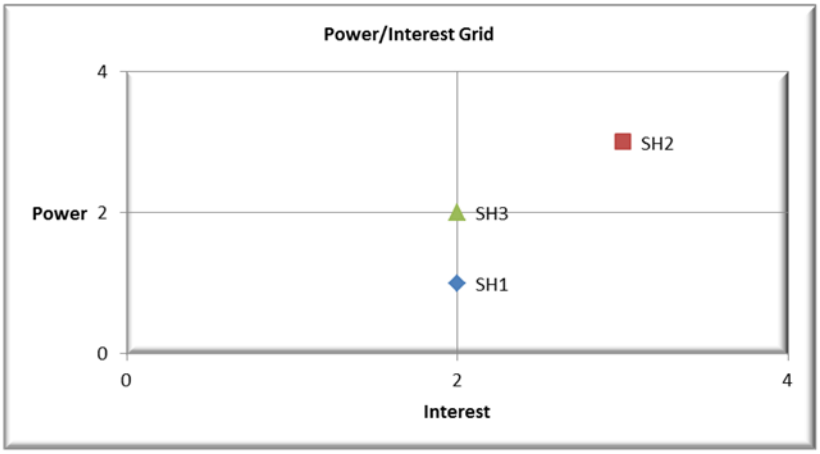
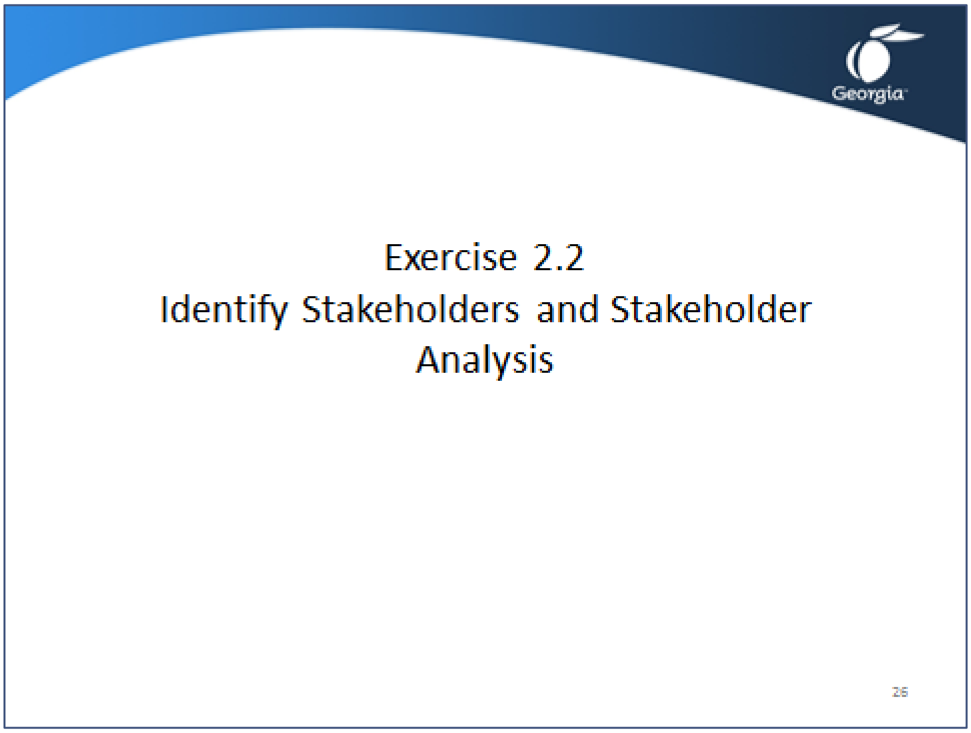


Figure based on the Project Management Institute, *A Guide to the Project Management Body of Knowledge, (PMBOK® Guide*)– Fifth Edition, Project Management Institute, Inc., 2013, Figure 13-4 Page 397.

[1] *PMBOK® Guide*, Page 395-396.

Exercise 2.2 Identify Stakeholders and Perform Stakeholder Analysis



Exercise 2.2
Identify Stakeholders and Stakeholder Analysis

26

Using the information provided in the Case Study and the templates below, determine the project stakeholders in a Stakeholder Register, then perform a Stakeholder Analysis.

Stakeholder Register

SH ID	Stakeholder Name	Project Role	Major Requirements	Main Expectations

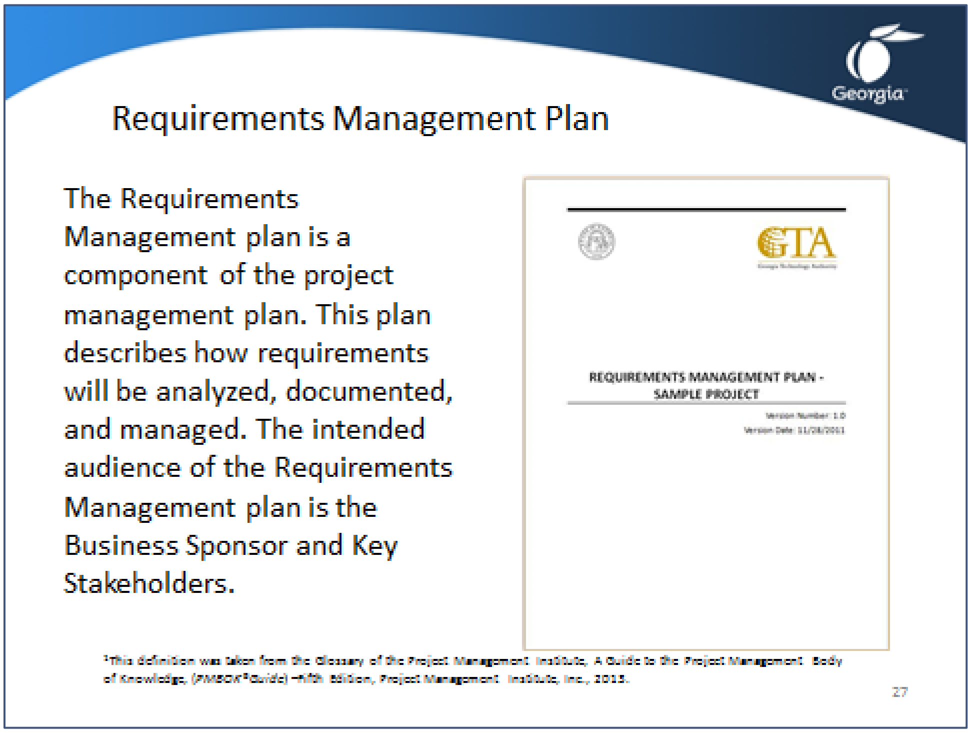
Stakeholder Analysis

SH ID	Stakeholder Name	Impact	Influence	Interest	Power	Urgency	Legitimacy

3= High
2= Medium
1= Low

Impact: The SH ability to effect changes to the project's execution or planning.
Influence: The SH active involvement in the project.
Interest: The SH level of interest regarding project outcomes.
Power: The SH ability to impose their will on the project.
Urgency: The SH need for immediate attention.
Legitimacy: The SH involvement is appropriate.

Topic 3: Developing the Requirements Management Plan



Georgia

Requirements Management Plan

The Requirements Management plan is a component of the project management plan. This plan describes how requirements will be analyzed, documented, and managed. The intended audience of the Requirements Management plan is the Business Sponsor and Key Stakeholders.

GTA

REQUIREMENTS MANAGEMENT PLAN - SAMPLE PROJECT

Version Number: 1.0

Version Date: 11/28/2011

[1]This definition was taken from the Glossary of the Project Management Institute, A Guide to the Project Management Body of Knowledge, (*PMBOK®Guide*) –Fifth Edition, Project Management Institute, Inc., 2013.

27

The Requirements Management plan is a component of the project management plan. This plan describes how requirements will be analyzed, documented, and managed. [1] The intended audience of the Requirements Management plan is the Business Sponsor and Key Stakeholders.

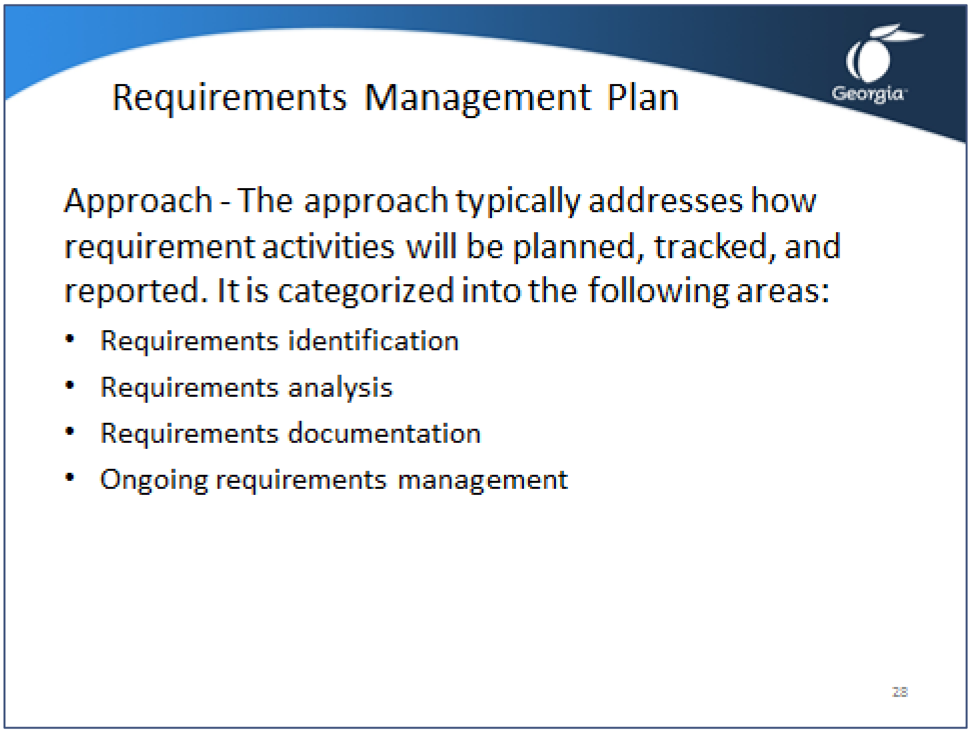
Example:
The purpose of the <Agency> Requirements Management Plan is to establish a common understanding of how requirements will be identified, analyzed, documented, and managed for the <Agency> fiber optic cable project.

Requirements will be divided into two categories: project requirements and product requirements. Project requirements are the requirements identified to meet the needs of the project and ensure its completion and readiness to hand over to operations. These consist mostly of non-technical requirements. Product requirements are the requirements identified to meet the technical specifications of the product being produced as a result of the project: the <Agency> fiber optic cable. These will consist of requirements to ensure that performance specifications are met, cable properties are properly documented, and manufacturing thresholds are identified and documented.

The inputs for the requirements management plan include the <Project Name> Project Charter and Stakeholder Register.

[1] This definition was taken from the Glossary of the Project Management Institute, A Guide to the Project Management Body of Knowledge, (*PMBOK®Guide*) –Fifth Edition, Project Management Institute, Inc., 2013.

Topic 3: Developing the Requirements Management Plan - Approach



Requirements Management Plan

Approach - The approach typically addresses how requirement activities will be planned, tracked, and reported. It is categorized into the following areas:

- Requirements identification
- Requirements analysis
- Requirements documentation
- Ongoing requirements management

28

Requirements Management Approach

The requirements management approach is the methodology the project team will use to identify, analyze, document, and manage the project's requirements. Many organizations use a standard approach for all projects, but based on the characteristics of each project, this approach may require some changes. The *PMBOK® Guide* defines this approach as "How requirements activities will be planned, tracked, and reported."

Example:

The approach we will use for requirements management for the BrightStar project will be broken down into four areas: requirements identification, requirements analysis, requirements documentation, and ongoing requirements management.

Requirements Identification

The <Project Name> project team will facilitate various methods to collect requirements which may include: interviews, focus groups, facilitated workshops, group creativity techniques, questionnaires and surveys, or product prototypes. These will be conducted among the project stakeholders to ensure all requirements are captured.

Requirements Analysis

The <Project Name> project team will analyze requirements to determine if they fall into project or product categories. Additionally, this analysis will determine where in the WBS the requirements will fall or what work activities correspond to particular requirements. Accountability and priority for each requirement will also be determined as part of the analysis. Finally, metrics and acceptance criteria must be determined for all requirements in order to provide a baseline for understanding when a requirement has been fulfilled to an acceptable level.

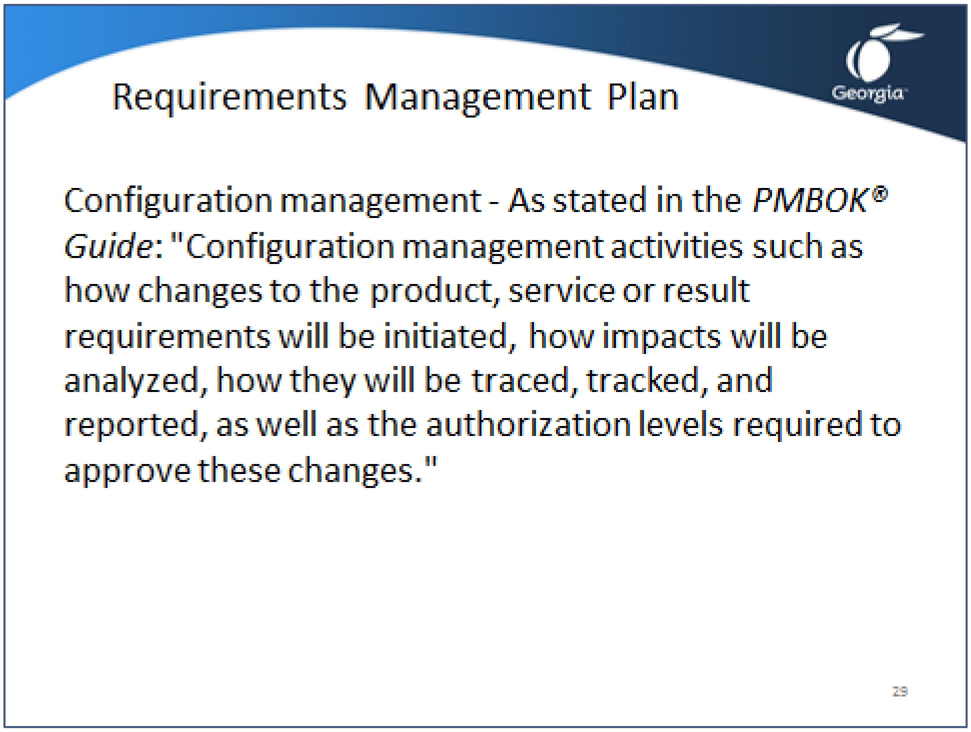
Requirements Documentation

Once requirements have been identified and analyzed, they will be documented and assigned to the responsible personnel. These requirements will be added to the <Project Name> project plan and the project team will determine what methodology the responsible personnel will use to track and report on the status of each requirement. All requirements will also be added to the project requirements checklist which must be completed before formal project closure is accepted by the project sponsor.

Ongoing Requirements Management

Throughout the project lifecycle, the project manager will ensure all team members are reporting requirement status and raising any issues or concerns with their assigned requirements as appropriate. As the project matures there may be situations in which requirements must change or be altered in some way. The project team must follow the established change control process in order to propose any changes to requirements and receive approval from the change control board. Ongoing requirements management also includes receiving approval of all requirements by all vested parties as part of project closure.

Topic 3: Developing the Requirements Management Plan – Configuration Management



Requirements Management Plan

Configuration management - As stated in the *PMBOK® Guide*: "Configuration management activities such as how changes to the product, service or result requirements will be initiated, how impacts will be analyzed, how they will be traced, tracked, and reported, as well as the authorization levels required to approve these changes."

29

Configuration Management

In order to effectively manage a project, communication must be managed and controlled. Additionally, every effort must be taken to identify requirements thoroughly during project initiation and planning. However, there are often situations which require changes to a project or its requirements. In these situations it is important to utilize configuration management to consider proposed changes, establish a process to review and approve any proposed changes, and to implement and communicate these changes to the stakeholders. As stated in the *PMBOK® Guide*: "Configuration management activities such as how changes to the product, service or result requirements will be initiated, how impacts will be analyzed, how they will be traced, tracked, and reported, as well as the authorization levels required to approve these changes."

Example:

For the <Project Name> project, the Requirements Management Plan will utilize the configuration management activities outlined in the Configuration Management Plan. Key items include documentation/version control and change control:

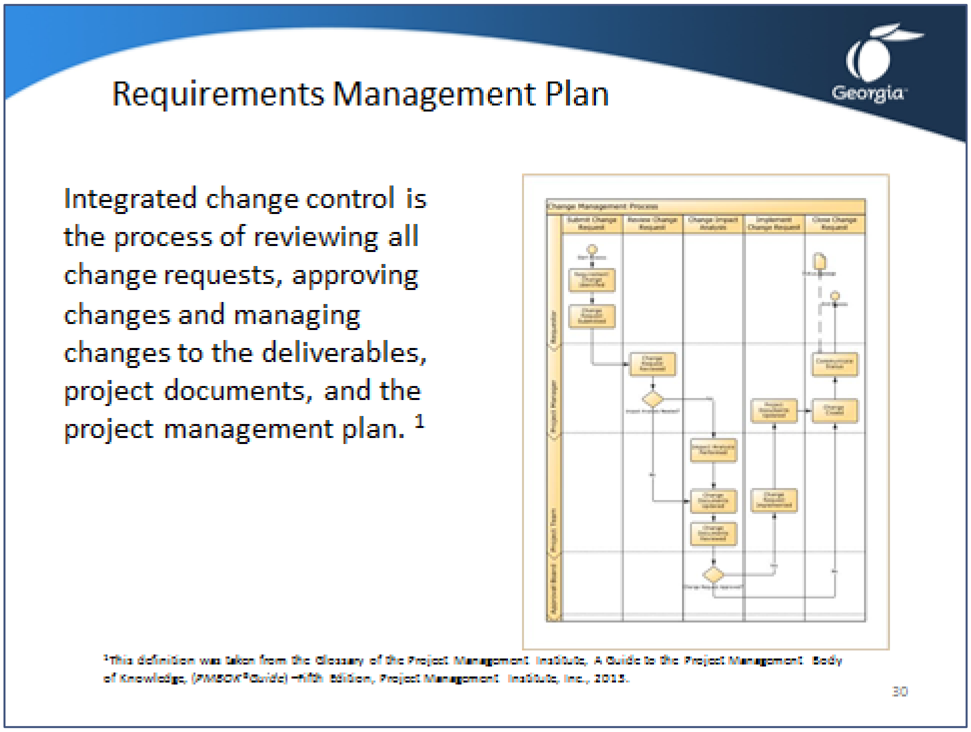
Documentation and Version Control

All project documentation will be loaded into the Configuration Management Database (CMDB) as the central repository for the <Project Name> project. Appropriate permissions will be granted to the project team for editing and revising documentation. **Any proposed changes to project requirements must be reviewed by the Configuration Control Board (CCB) and have written approval by the project sponsor before any documentation changes are made.** Once these proposed changes are approved and the documentation is edited, the project manager will be responsible for communicating the change to all project stakeholders.

Change Control

Any proposed changes in project requirements must be carefully considered before approval and implementation. Such changes are likely to impact project scope, time, and/or cost, perhaps significantly. Any proposed changes to project requirements will be reviewed by the CCB. The role of the CCB is to determine the impact of the proposed change on the project, seek clarification on proposed change, and ensure any approved changes are added to the CMDB. The project sponsor, who also sits on the CCB, is responsible for approving any changes in project scope, time, or cost and is an integral part of the change review and approval process.

Topic 3: Developing the Requirements Management Plan – Integrated Change Control Process



Requirements Management Plan

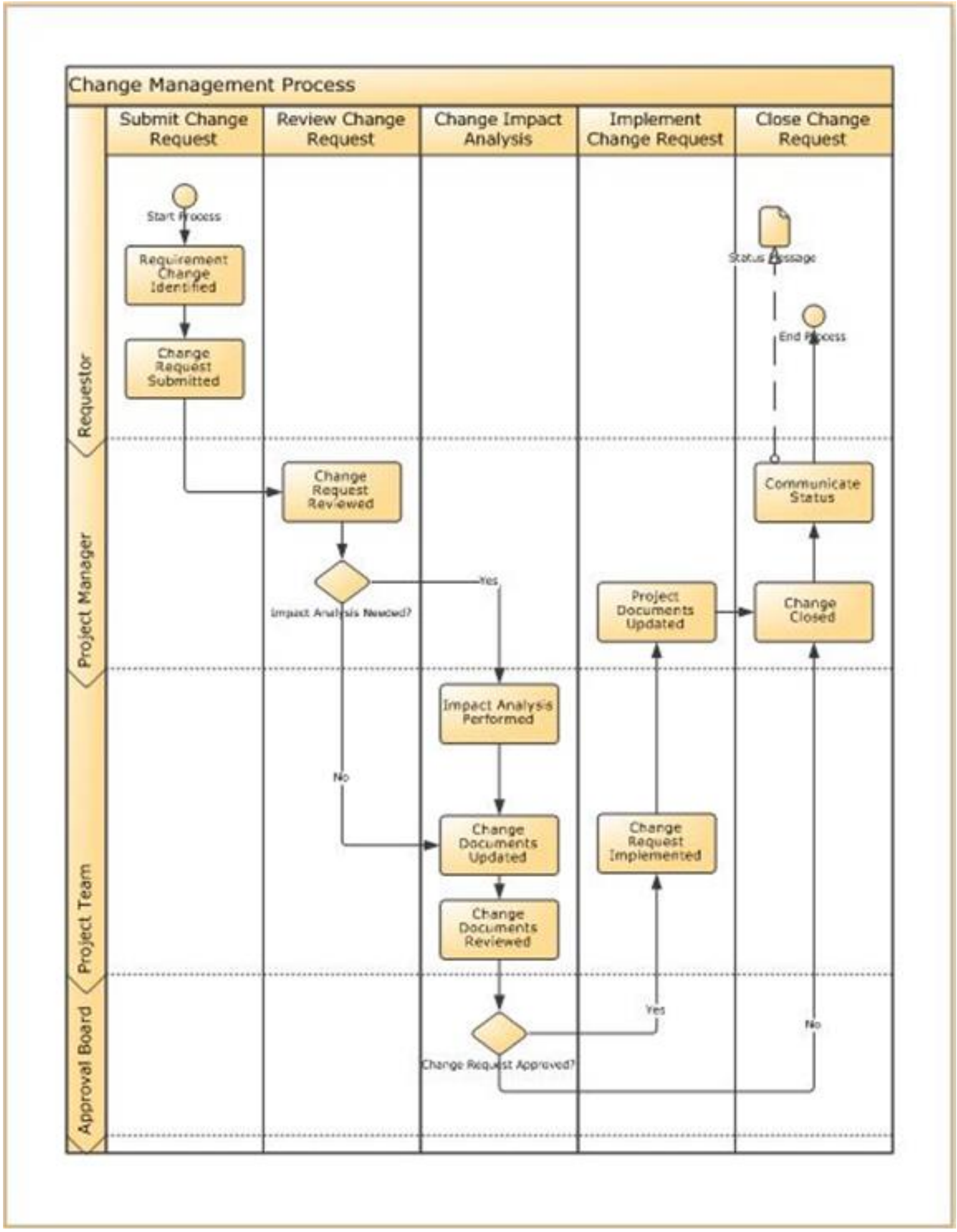
Integrated change control is the process of reviewing all change requests, approving changes and managing changes to the deliverables, project documents, and the project management plan. [1]

[1] This definition was taken from the Glossary of the Project Management Institute, A Guide to the Project Management Body of Knowledge, (*PMBOK®Guide*) –Fifth Edition, Project Management Institute, Inc., 2013.

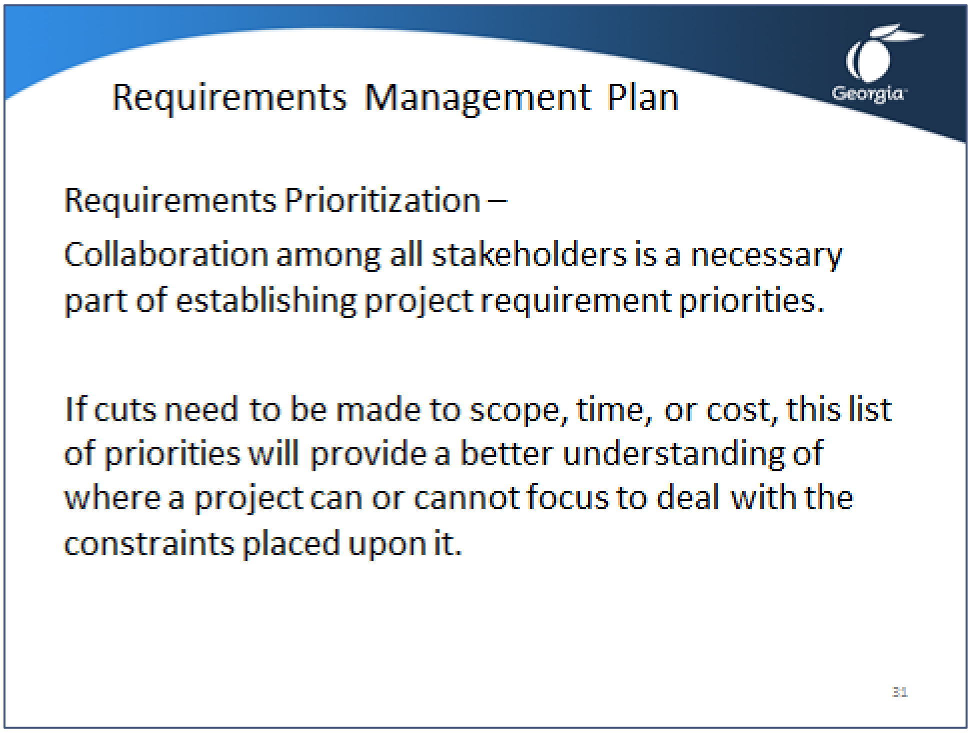
30

This process facilitates a proactive approach to Change Management by logically defining a set of activities that support the request, approval and implementation of changes to the project. The five steps of the change management process are: Submit a Request, Review the Request, Perform Impact Analysis, Implement the Change, and Close the Request. The output of this process is a change request form as well as a change log. The change request form incorporates the change description and impact information and is used to guide decision making as well as project planning. The change log serves as a control document that can be used to monitor the progress and status of the change request.

Integrated Change Control Process



Topic 3: Developing the Requirements Management Plan – Prioritization



Requirements Management Plan

Requirements Prioritization –

Collaboration among all stakeholders is a necessary part of establishing project requirement priorities.

If cuts need to be made to scope, time, or cost, this list of priorities will provide a better understanding of where a project can or cannot focus to deal with the constraints placed upon it.

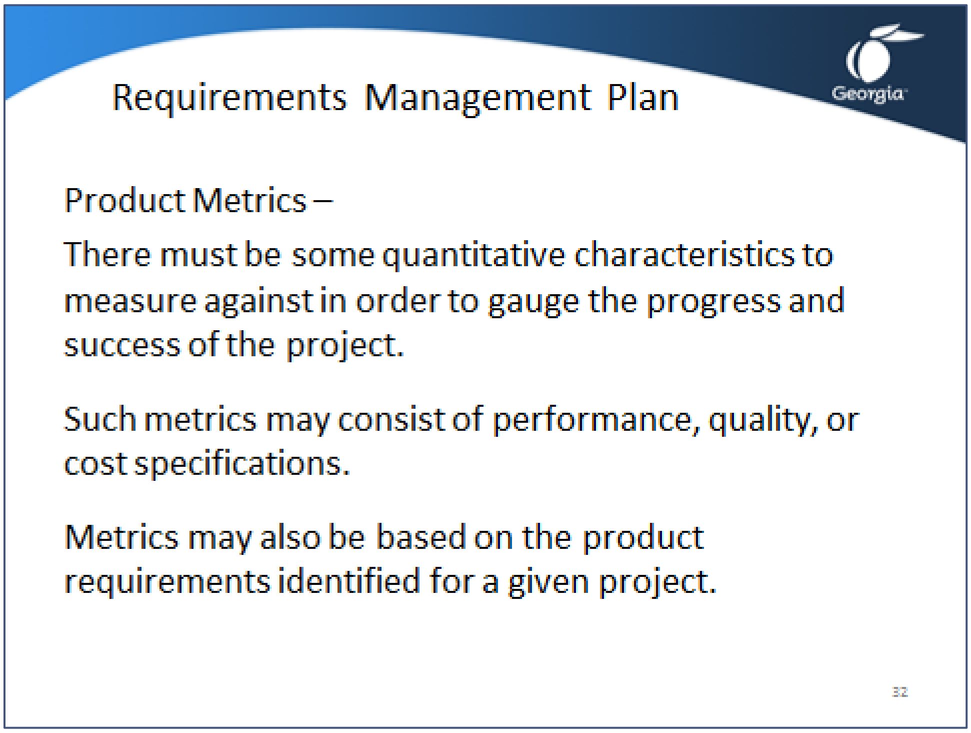
Requirements Prioritization Process
Prioritizing requirements is an important part of requirements management. Developers or service providers do not always know what requirements are most important to a customer. Conversely, customers do not always understand the scope, time, and cost impacts of their requirements on a project. Collaboration among all stakeholders is a necessary part of establishing project requirement priorities. If cuts need to be made to scope, time, or cost, this list of priorities will provide a better understanding of where a project can or cannot focus to deal with the constraints placed upon it. One way to do this is to group requirements into priority categories such as high, medium, and low priority based upon the importance of the requirement. There may be hundreds of requirements in a large project so this type of categorically-based method would be helpful. NOTE: there are many methods by which priorities are determined and these should be explored based on the size and complexity of the project.

Example:
The <Project Name> project manager will facilitate stakeholder meetings in order to establish priorities for all project requirements. This project will use a three-level scale in order to prioritize requirements. The chart below illustrates these levels and defines how requirements will be grouped:

Priority Level	Definition
High	These requirements are mission critical. They are required for project/product success or for progression to the next project phase.
Medium	These requirements support product/process operations but can be completed under the product release.
Low	These requirements are quality and/or functional enhancements and are desirable only if time and resources permit.

As the project moves forward and additional constraints are identified or there are issues with resources, it may be necessary for the project team and stakeholders to meet in order to determine what requirements must be achieved, which can be re-baselined, or which can be omitted. These determinations will be made in a collaborative effort based on the priorities of the requirements and which level they are assigned in accordance with the chart above. As any changes in requirements are made, all project documentation must be updated in the CMDB and communicated to all project stakeholders.

Topic 3: Developing the Requirements Management Plan – Product Metrics



Requirements Management Plan

Product Metrics –

There must be some quantitative characteristics to measure against in order to gauge the progress and success of the project.

Such metrics may consist of performance, quality, or cost specifications.

Metrics may also be based on the product requirements identified for a given project.

32

Product Metrics
Product metrics are an important part of determining a project's success. There must be some quantitative characteristics to measure against in order to gauge the progress and success of the project. Product metrics are usually technical in nature though not always. Such metrics may consist of performance, quality, or cost specifications. Metrics may also be based on the product requirements identified for a given project.

Example:
Product metrics for the <Project Name> project will be based on cost, quality, and performance requirements as outlined in the project charter. In order to achieve project success, the <Project Name> product must meet or exceed all established metrics.

Cost:

- <Project Name> cable product must cost less than $6,000 per linear kilometer for fiber counts of 12-72 fibers; less than $8,000 per linear kilometer for fiber counts of 84-180 fibers; less than $10,000 per linear kilometer for fiber counts of 192-288 fibers.

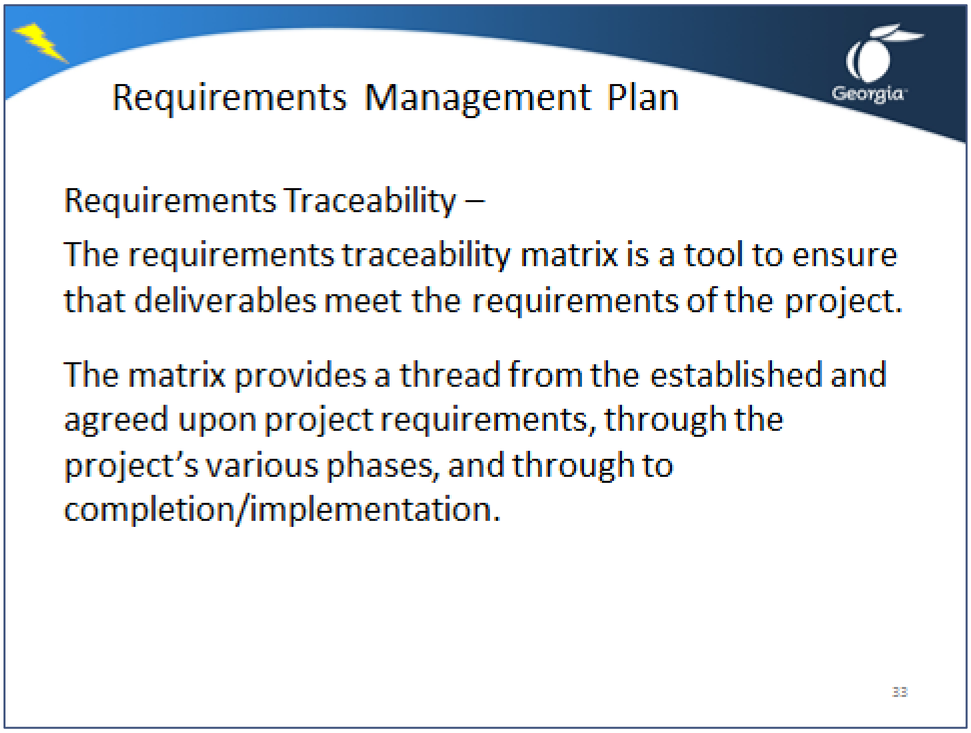
Quality:

- <Project Name> cable product must achieve less than 10% attenuation in temperature cycle testing
- <Project Name> cable product must achieve a minimum bending radius of less than 10 feet
- <Project Name> cable product must weigh less than 1.0 lb. per linear foot for fiber counts of 12-180 fibers and less than 2.0 lbs. for fiber counts greater than 180

Performance:

- <Project Name> cable must achieve an average attenuation of less than 0.1% per linear kilometer at 1550nm
- <Project Name> cable must achieve an average attenuation of less than 0.5% per linear kilometer at 1610nm
- <Project Name> cable must have a diameter of less than 1.0” for 12-72 fiber cables; less than 1.5” for 84-180 fiber cables; and less than 2.0” for 192-288 fiber cables

Topic 3: Developing the Requirements Management Plan – Traceability



Requirements Management Plan

Requirements Traceability –

The requirements traceability matrix is a tool to ensure that deliverables meet the requirements of the project.

The matrix provides a thread from the established and agreed upon project requirements, through the project’s various phases, and through to completion/implementation.

33

Requirements Traceability
The requirements traceability matrix is a tool to ensure that deliverables meet the requirements of the project. The matrix provides a thread from the established and agreed upon project requirements, through the project’s various phases, and through to completion/implementation. This ensures that the product specifications and features satisfy the requirements on which they were based. Any interim project tasks associated with the requirements should be included. An example of this is test cases which will utilize metrics, based on the product requirements, which are linked to the project charter and design document. This allows a team member or stakeholder to follow a requirement through all tasks required for its completion.

Example:
Below is the requirements traceability matrix for the <Project Name> project. The purpose of the requirements traceability matrix is to ensure all product requirements are completed in accordance with the project charter. This matrix provides a thread from all product requirements through design, testing, and user acceptance. Design document and charter references are contained in the <Project Name> Project Configuration Management Plan. Any approved changes in project scope or requirements will result in changes to the traceability matrix below. Based on impacts of the approved changes, the Project Manager will make the necessary changes to the matrix and communicate those changes to all project stakeholders.

Requirements Traceability Matrix	
Project Name:	Sample Project
Project Description:	
Cost Center:	Research and Development

ID	Requirements Description	Use Case ID	Use Case Name	Test Case ID	Test Case Name	Date Tested
R1	Reduce cable building cost per linear foot by 15%	UC0001		TC0001		
R2	Improve attenuation in temperature testing by 10%	UC0002		TC0002		
R3	Improve fiber cable bending radius by 10%	UC0003		TC0003		
R4	Reduce fiber cable weight by 10%	UC0004		TC0004		

Lesson 2 Summary: Learning Objectives Recap

- **Explain how customer needs are identified and developed into requirements**
  Needs are the fundamental driving force behind projects. The emergence of a need sets off the whole project process. Customer needs evolve from something very vague to something well-structured and clearly understood.
  There are three phases in the Needs Lifecycle:
  - Needs Emergence
  - Needs Recognition
  - Needs Articulation
- **Describe the role of the stakeholder in determining requirements**
  Stakeholders are comprised of customers, sponsors, the performing organization, and the public who are actively involved in the project, or whose interests may be positively or negatively affected by the execution or completion of the project. They may also exert influence over the project and its deliverables. Stakeholders may be at different levels of the organization and may possess different authority levels, or may be external to the performing organization of the project.
- **Discuss the components of the Requirements Management Plan**
  The Requirements Management plan is a component of the project management plan. This plan describes how requirements will be analyzed, documented, and managed. The intended audience of the Requirements Management plan is the Business Sponsor and Key Stakeholders. The major components of the Requirements Management Plan are:

  - Requirements Management Approach
  - Configuration Management
  - Requirements Prioritization Process
  - Product Metrics
  - Requirements Traceability

Notes

LESSON 3: DEFINING PROJECT REQUIREMENTS

Topic 1: Core Components of Requirements

Topic 2: Understanding Product Scope

Topic 3: Developing Functional Requirements

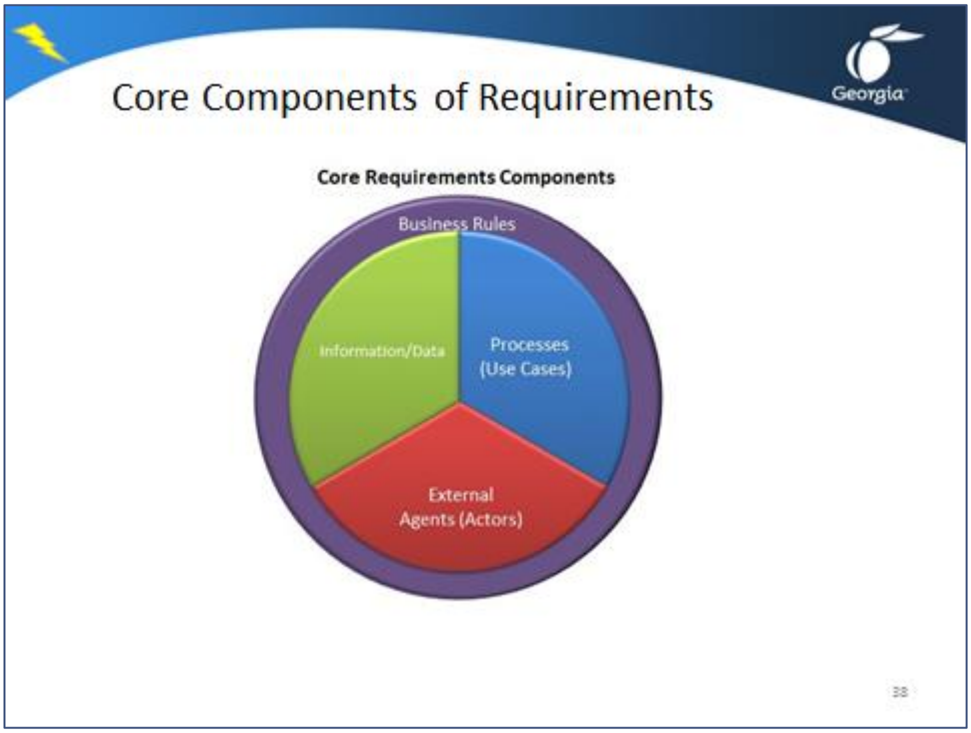
Student Learning Objectives

After completing this lesson you should be able to

- Recognize core components of requirements
- Explain the components and purpose of Product Scope
- Describe how to develop and communicate Functional requirements

Approximate Presentation time: 3 hours

Topic 1: Core Components of Requirements



Core Components of Requirements

There are several core components of requirements as described below:

DATA (Entities and Attributes)

An **entity** is a uniquely identifiable person, thing, or concept whose information is important to the business. They are named by using nouns or a noun phrase. Entities are used to create database tables to store the information important to the business
Example:
INSTRUCTOR – A person who teaches a CLASS SESSION.
STUDENT – An employee who registers for a CLASS SESSION
CLASS SESSION – This is a scheduled training session that will cover a particular COURSE topic
COURSE – A program of study that includes course materials, learning objectives, student exercises, and instructor evaluations.

An **attribute** is a piece of information that is a characteristic of an entity. They are named with nouns or noun phrases with adjectives.
Example:
Student Last Name
Student First Name
Instructor Number

Each attribute is further described by three characteristics. These detail the data requirements and help identify other entities and attributes.

UNIQUENESS: Is the Student First Name unique?
OPTIONAL vs. MANDATORY: Is “First Name” optional or mandatory for a student?
REPITITIONS: How many “First Names” can a student have? How many phone numbers can

a student have?

PROCESSES or USE CASES

A **process** is a business activity that transforms inputs (entities and attributes) into outputs. A Use Case is a high level process that will be included in the software automation. Processes and Use Cases are named with a verb phrase (verb and noun).

Example:
Schedule Class
Register Student
Assign Instructor

EXTERNAL AGENTS or ACTORS

An **external agent** is a person, organization, or system with which the business area interacts. An **actor** is any resource that uses software automation. They are always external to the system. External Agents and Actors are named with the title of the person, organization, or system that is represented.

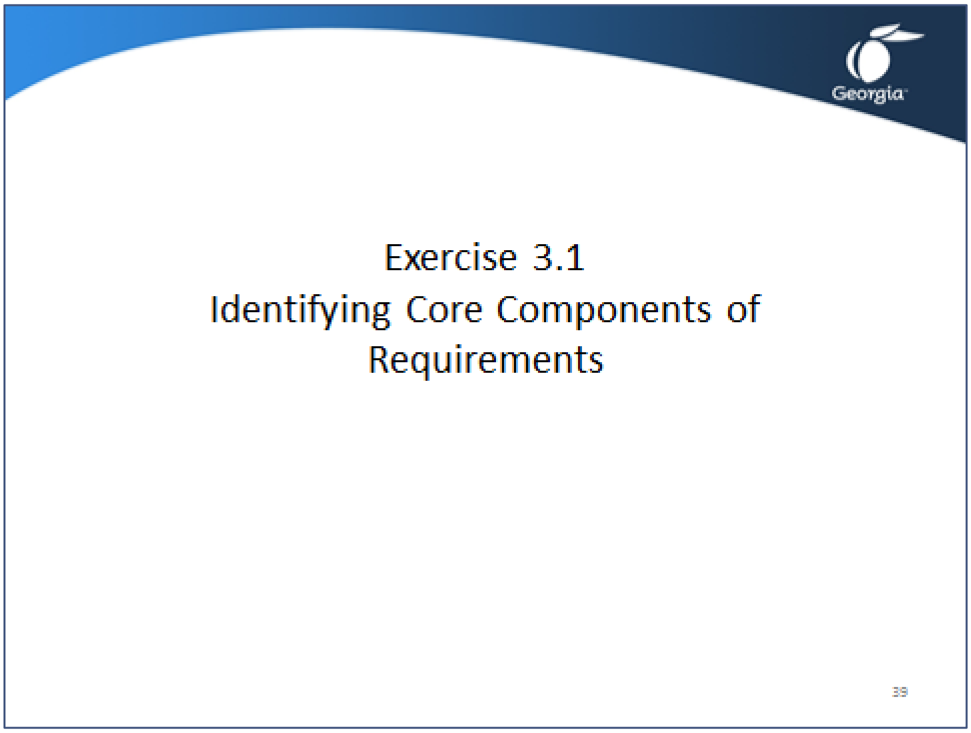
Example:
Training Administrator
Student
Conference Room Management System

BUSINESS RULES

A **business rule** is a condition that governs the way work is done (a constraint on the business). Business Rules are named with a descriptive sentence.

Example:
A student may register for more than one class at a time.
Every class must be taught by one and only one instructor.
A class can include at most 15 students.

Exercise 3.1 Identifying Core Components of Requirements



Exercise 3.1
Identifying Core Components of Requirements

39

Using the information provided in the Case Study identify the core components of the requirements.

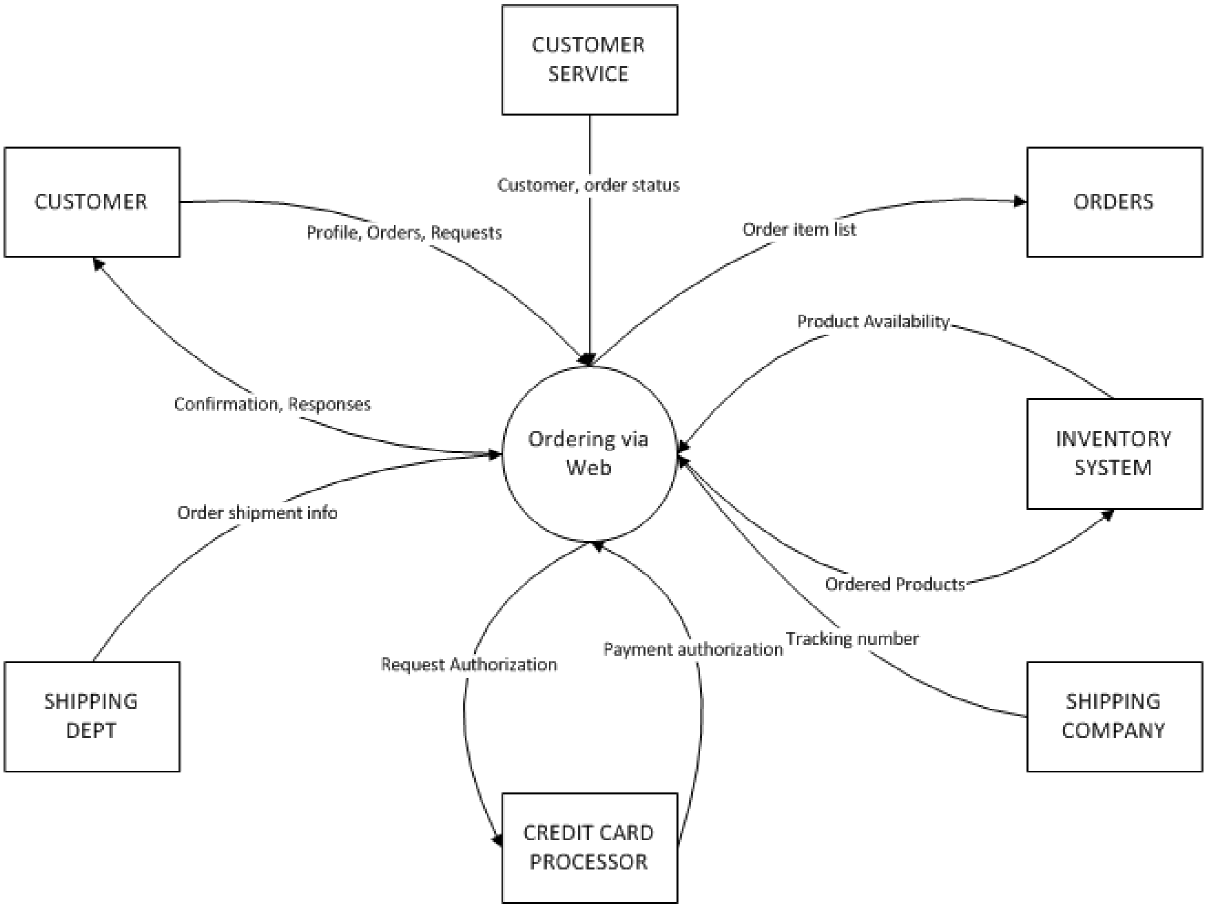
ENTITIES/ATTRIBUTES (DATA):

PROCESSES:

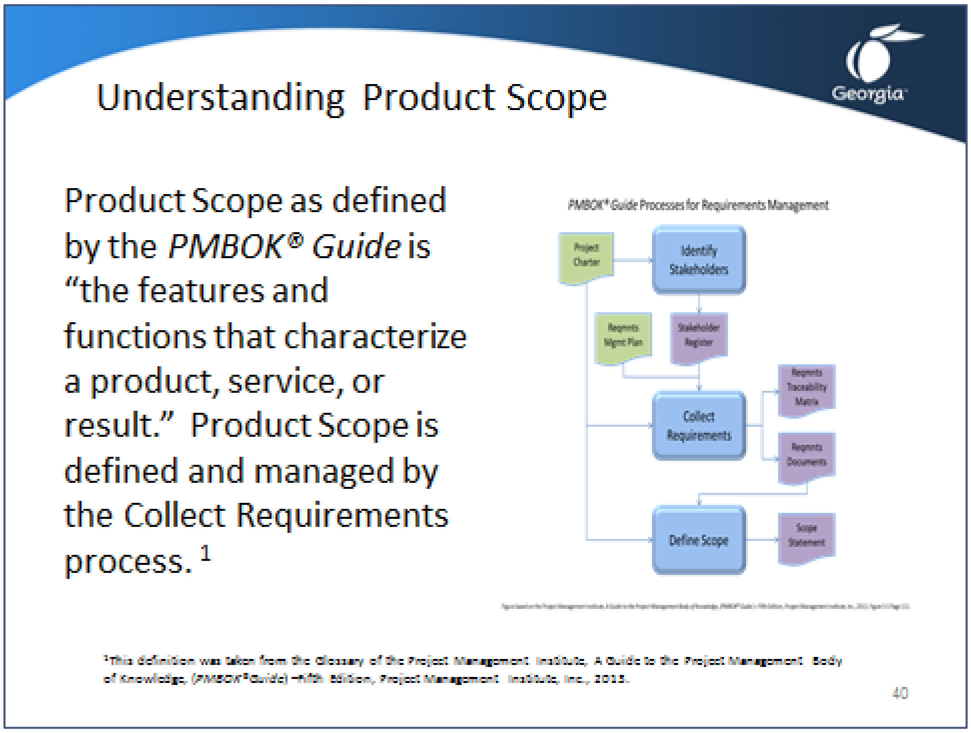
BUSINESS RULES:

EXTERNAL AGENTS or ACTORS:

Context Level Dataflow Diagram



Topic 2: Understanding Product Scope



Understanding Product Scope

Product Scope as defined by the *PMBOK® Guide* is "the features and functions that characterize a product, service, or result." Product Scope is defined and managed by the Collect Requirements process. [1]

PMBOK® Guide Processes for Requirements Management

[illegible]

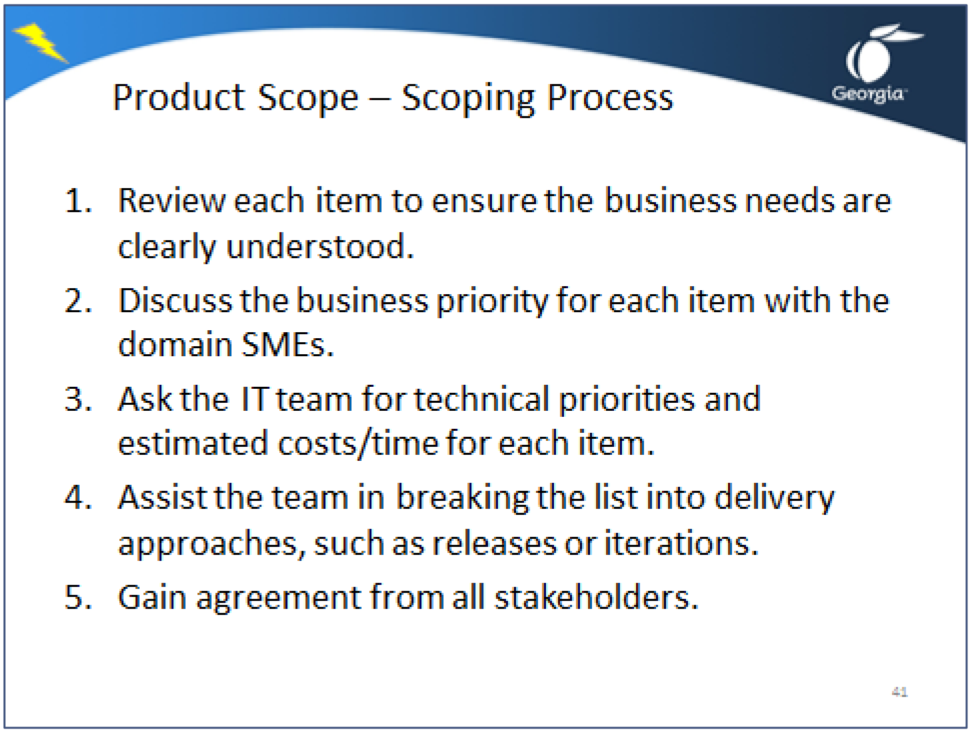
[1]This definition was taken from the Glossary of the Project Management Institute, A Guide to the Project Management Body of Knowledge, (*PMBOK®Guide*) –Fifth Edition, Project Management Institute, Inc., 2013.

40

Product scope as defined by the *PMBOK® Guide* is "the features and functions that characterize a product, service, or result." [1] Product scope is defined and managed by the Collect Requirements process. Now that the Project Charter, Requirements Management Plan, and Stakeholder Register are complete we can begin to define the features and functions that will meet the objectives of our product, service, or result. After the Collect Requirements process is completed the project manager will then produce the Project Scope Statement through the Define Scope process. The Project Scope Statement and the collected requirements form the basis for the Work Breakdown Structure.

[1] This definition was taken from the Glossary of the Project Management Institute, A Guide to the Project Management Body of Knowledge, (*PMBOK®Guide*) –Fifth Edition, Project Management Institute, Inc., 2013.

Topic 2: Understanding Product Scope – Five Step Scoping Process



Product Scope – Scoping Process

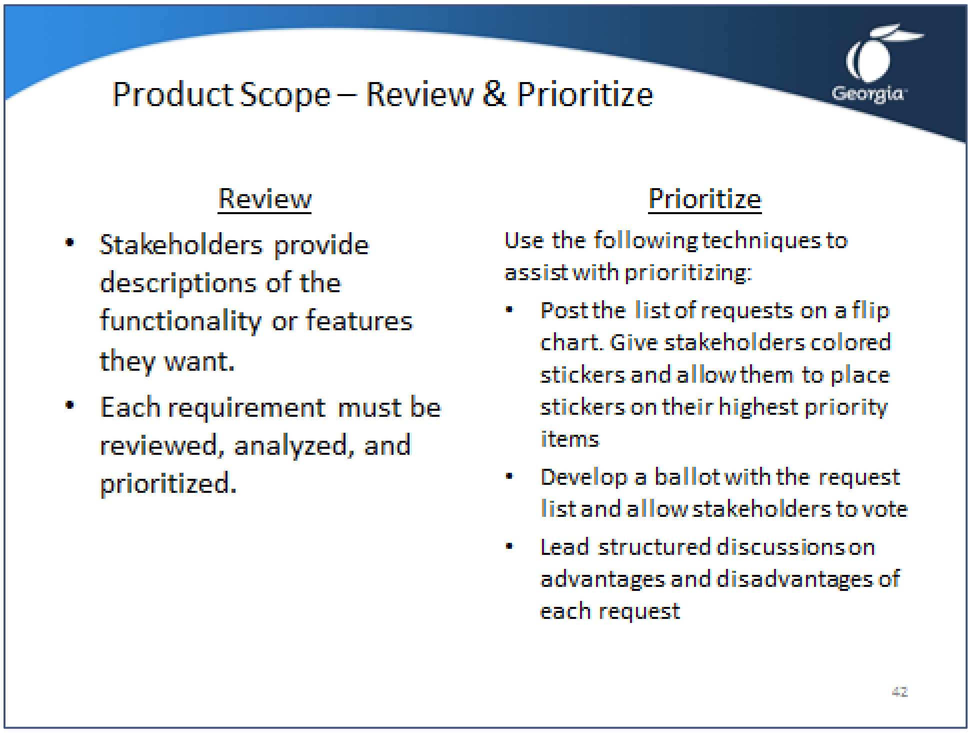
1. Review each item to ensure the business needs are clearly understood.
2. Discuss the business priority for each item with the domain SMEs.
3. Ask the IT team for technical priorities and estimated costs/time for each item.
4. Assist the team in breaking the list into delivery approaches, such as releases or iterations.
5. Gain agreement from all stakeholders.

41

Determining the product scope involves prioritization and consensus. The team may use the following steps iteratively as needed to move from business requests to the initial solution plan, which will also serve as a basis for the product scope. This will be updated throughout the project as requirements are refined.

1. Review/discuss each requirement request to ensure the business needs are clearly understood in order to estimate effectively.
2. Discuss the business priority for each item with the domain SMEs
3. Ask the IT team for technical priorities and estimated costs/time for each item
4. Assist the team in breaking the list into delivery approaches, such as releases or iterations, based on priorities and budget constraints
5. Gain agreement from all stakeholders.

Topic 2: Understanding Product Scope – Step 1&2 – Review/Prioritize



Product Scope – Review & Prioritize

Review

- Stakeholders provide descriptions of the functionality or features they want.
- Each requirement must be reviewed, analyzed, and prioritized.

Prioritize

Use the following techniques to assist with prioritizing:

- Post the list of requests on a flip chart. Give stakeholders colored stickers and allow them to place stickers on their highest priority items
- Develop a ballot with the request list and allow stakeholders to vote
- Lead structured discussions on advantages and disadvantages of each request

42

Stakeholders provide descriptions of the functionality or features they want. Each requirement must be reviewed, analyzed, prioritized, estimated, and allocated to a release or iteration. A table like the one below can be used to organize this information into an initial solution plan.

Unique Number	Description (i.e. feature or function)	Business Priority	Technical Priority (H,M,L)	Estimate (cost, time)	Phase, release, iteration
1	Allow customers to place orders online				
2	Allow customers to request items not in the catalog				
3	Add a wish list for customers to maintain future desired orders				
4	Build a chat function to answer customer questions online				
5	Add an option to request a catalog				

The business stakeholders should indicate the business priority for each requirement. To assist with prioritization, review the project charter and stakeholder register/analysis documents.

It is also helpful to ask which requirements such as business process:

- Are causing the most problems for the business
- Are the most critical to the business area
- Have the most risk associated with them
- Are costing the organization a significant amount of money
- Require a large number of resources

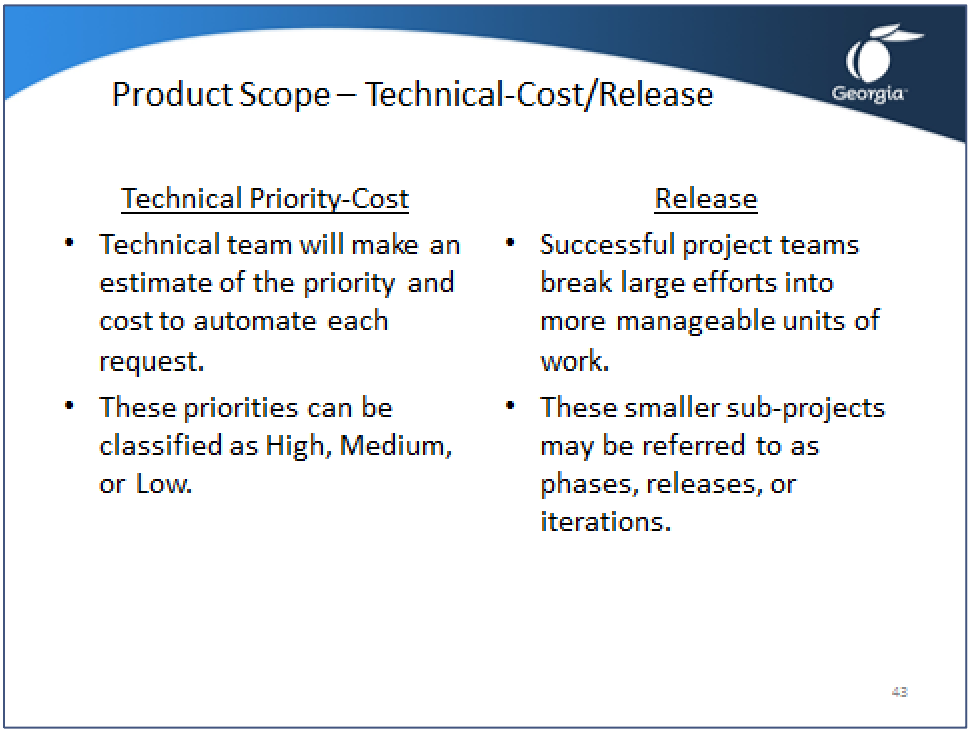
The following techniques can assist the stakeholders prioritize the requirements.

- Post the list of requests on a flip chart. Give stakeholders colored stickers and allow them to place stickers on their highest priority items
- Develop a ballot with the request list and allow stakeholders to vote
- Lead structured discussions on advantages and disadvantages of each request

A commonly used classification of requirements is the "MoSCoW" method.
M – Must Have
S- Should have
C – Could have
W – Won't have

Unique Number	Description (i.e. feature or function)	Business Priority	Technical Priority	Estimate (cost, time)	Phase, release, iteration
1	Allow customers to place orders online	M			
2	Allow customers to request items not in the catalog	C			
3	Add a wish list for customers to maintain future desired orders	S			
4	Build a chat function to answer customer questions online	S			
5	Add an option to request a catalog	C			

Topic 2: Understanding Product Scope – Step 3&4 – Technical-Cost/Release



Product Scope – Technical-Cost/Release

Technical Priority-Cost

- Technical team will make an estimate of the priority and cost to automate each request.
- These priorities can be classified as High, Medium, or Low.

Release

- Successful project teams break large efforts into more manageable units of work.
- These smaller sub-projects may be referred to as phases, releases, or iterations.

43

Once the technical team understands the requirements and the desired functional design, they can make an estimate of the cost to automate each one. These priorities can be classified as High, Medium, or Low since actual estimates may be difficult to determine at this early stage of the project.

Unique Number	Description (i.e. feature or function)	Business Priority	Technical Priority	Estimate (cost, time)	Phase, release, iteration
1	Allow customers to place orders online	M	H	H	
2	Allow customers to request items not in the catalog	C	L	M	
3	Add a wish list for customers to maintain future desired orders	S	M	M	
4	Build a chat function to answer customer questions online	S	M	M	
5	Add an option to request a catalog	C	L	L	

Successful project teams break large efforts into more manageable units of work. These smaller sub-projects may be referred to as phases, releases, or iterations.

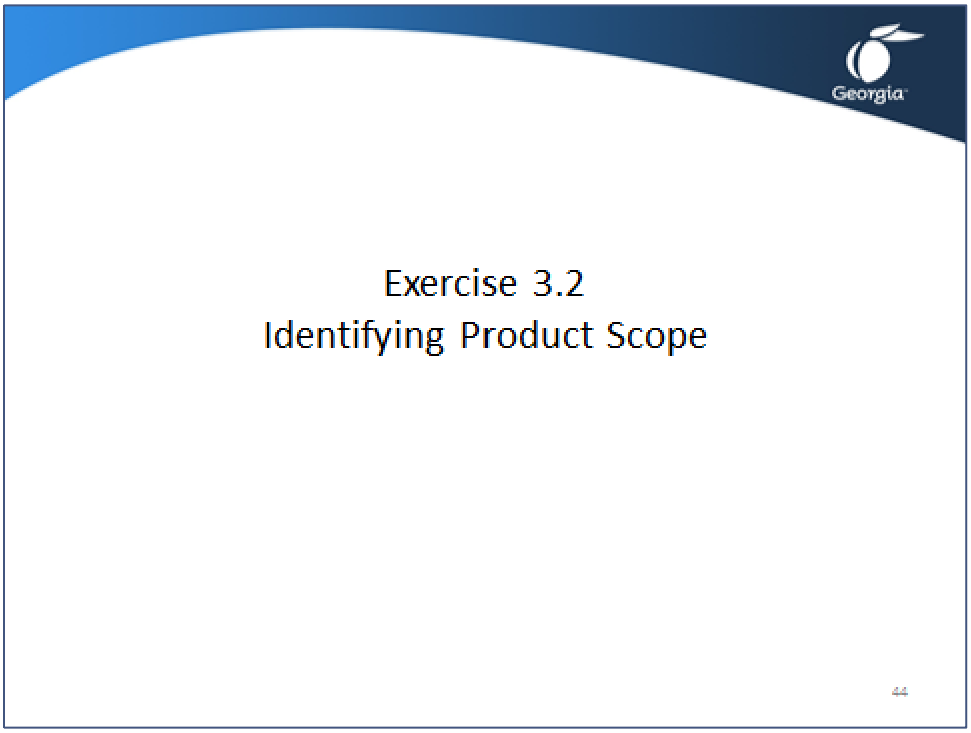
- Phase – a sub-set of the project
- Release – describes the components of the product which will be installed into production together
- Iteration – describes one pass through the product.

Unique Number	Description (i.e. feature or function)	Business Priority	Technical Priority	Estimate (cost, time)	Phase, release, iteration
1	Allow customers to place orders online	M	H	H	P1
2	Allow customers to request items not in the catalog	C	L	M	R1
3	Add a wish list for customers to maintain future desired orders	S	M	M	P1
4	Build a chat function to answer customer questions online	S	M	M	P1
5	Add an option to request a catalog	C	L	L	R1

The final step is getting approval from the stakeholders.

Reviewer/Approver	Rvwr	Apprv	Signature	Date
John Doe, Executive Sponsor		X		
Jan Jones, Customer Service		X		
Jeff Smith, Accounting	X			

Exercise 3.2 Identifying Product Scope



Georgia

Exercise 3.2
Identifying Product Scope

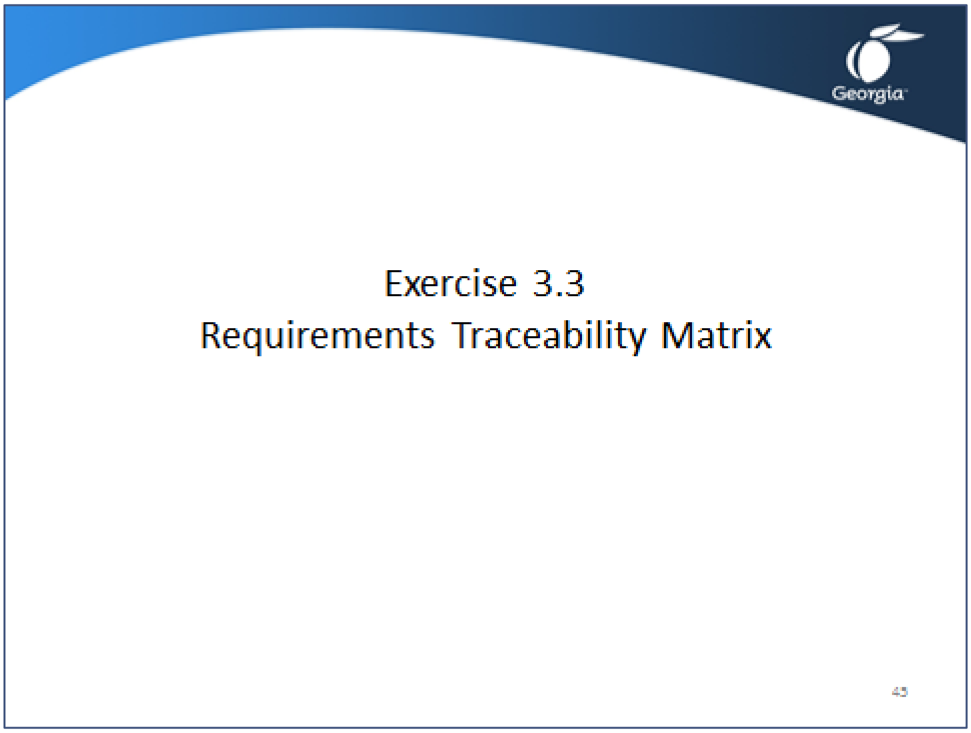
44

Using the information provided in the Case Study and the table on the following page identify the product scope.

Product Scope Planning Worksheet

Unique Number	Description (i.e. feature or function)	Business Priority	Technical Priority	Estimate (cost, time)	Phase, release, iteration

Exercise 3.3 Requirements Traceability Matrix



Exercise 3.3
Requirements Traceability Matrix

45

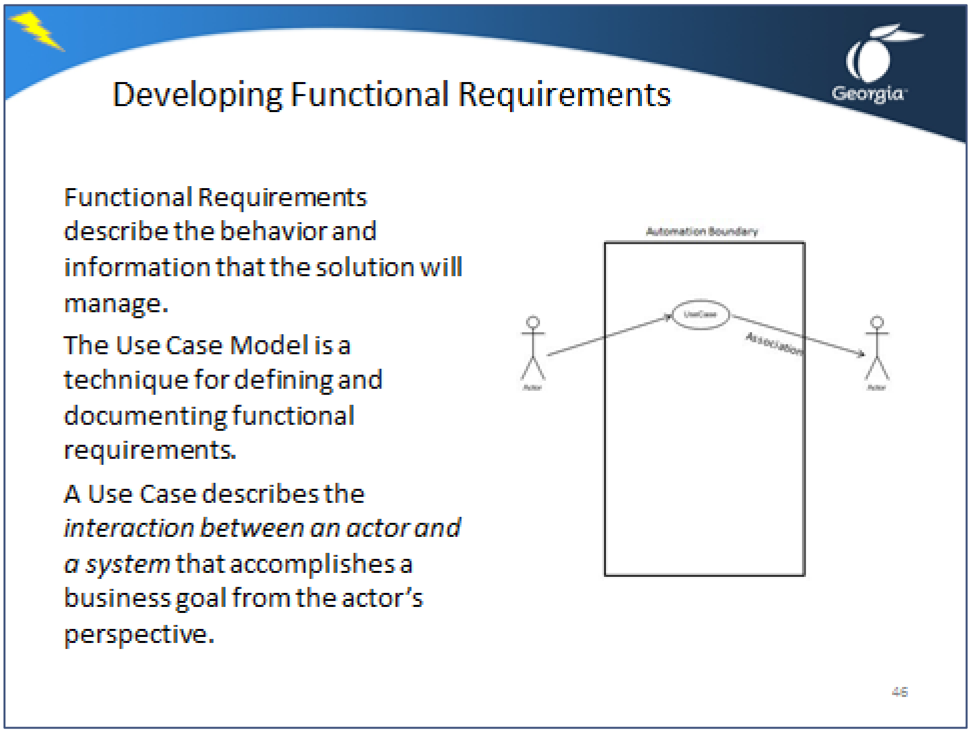
Using the information provided in the Case Study and the prior exercise begin to fill in the Requirements Traceability Matrix using the template below.

Requirements Traceability Matrix

Requirements Traceability Matrix	
Project Name:	Speedy Office Supplies Web Expansion Project
Project Description:	Create the retail shopping experience for customers on our website.
Cost Center:	Customer Service Department

ID	Requirements Description	Use Case ID	Use Case Name	Test Case ID	Test Case Name	Date Tested	Approved By
R1							
R2							
R3							
R4							
R5							
R6							
R7							

Topic 3: Developing Functional Requirements



Developing Functional Requirements

Functional Requirements describe the behavior and information that the solution will manage.

The Use Case Model is a technique for defining and documenting functional requirements.

A Use Case describes the *interaction between an actor and a system* that accomplishes a business goal from the actor's perspective.

Georgia
Automation Boundary
UseCase
Association
Actor
Actor
46

"Functional Requirements describe the behavior and information that the solution will manage." BABOK® v2.0

There are many techniques for defining and documenting functional requirements. This course will discuss Use Cases.

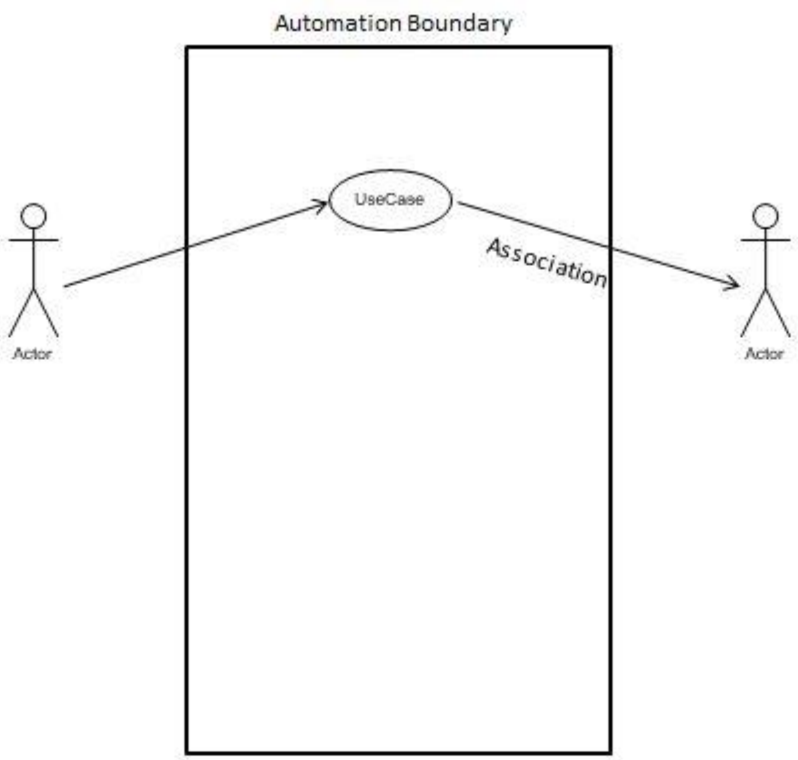
What is a Use Case? A Use Case describes the *interaction between an actor and a system* that accomplishes a business goal from the actor's perspective.

Detailing use cases is a technique used to document system functionality at an appropriate level of detail for the implementation team.

Components of a use case diagram include:

- Actors – drawn as a stick figure. An actor is the resource that will interact with the system.
- Automation Boundary – drawn as a large rectangle, it represents the system under discussion.
- Associations – A line from an actor to a use case shows that the actor is involved with the use case. A use case may have associations with many actors.
- Use Cases – Represent the goal of the actor, and are drawn as small ovals inside the automation boundary. They are also given a descriptive name.

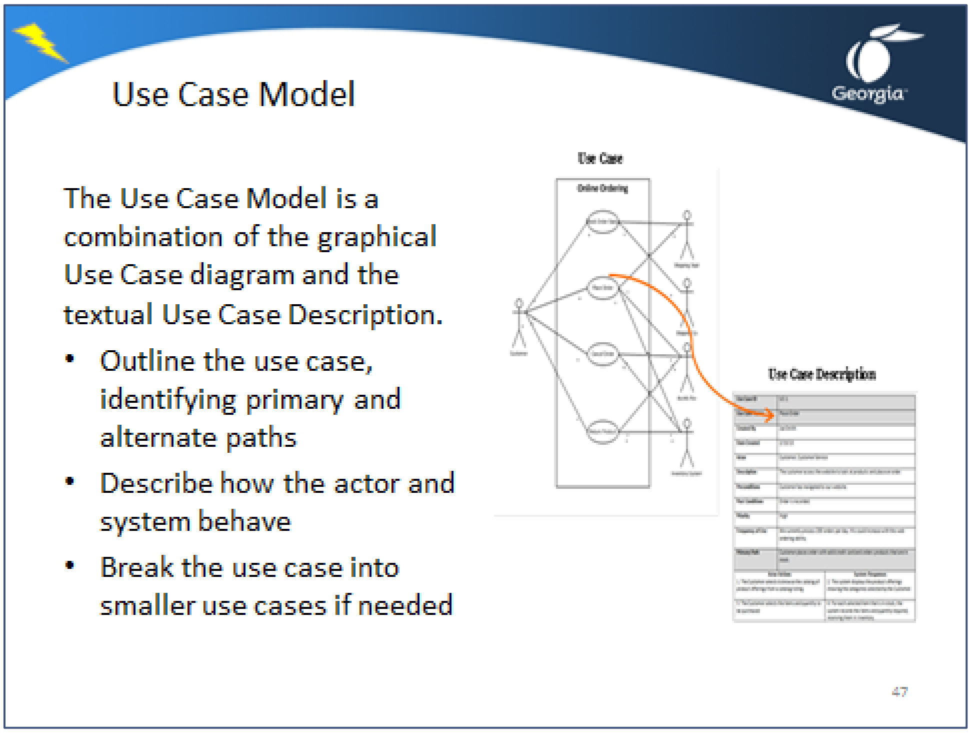
The diagram below represents a Use Case Diagram:



Developing detailed descriptions of steps or actions between a user (or actor) and a system to accomplish a goal will:

- Demonstrate the steps in a business scenario that will be carried out with software
- Help the team and the user think through how the system should operate
- Help the team and the user think about how the system should behave when problems occur
- Help the implementation team estimate effort
- Give the implementation team specific instructions about how the software should work under different conditions
- Be used by the implementation team to develop a custom solution
- Provide guidelines for the testers about what to expect from the system

Topic 3: Developing Functional Requirements – Use Cases



Use Case Model

The Use Case Model is a combination of the graphical Use Case diagram and the textual Use Case Description.

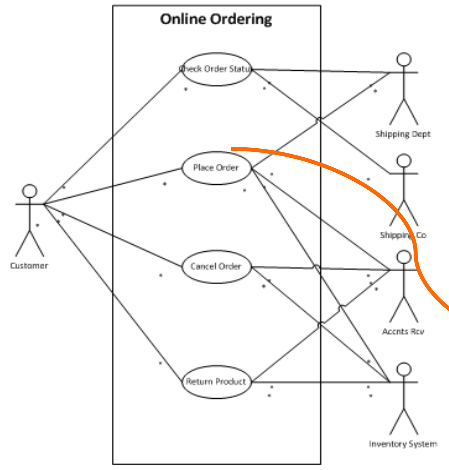
- Outline the use case, identifying primary and alternate paths
- Describe how the actor and system behave
- Break the use case into smaller use cases if needed

Each use case identified on the initial use case diagram represents a goal of the actor – each goal must be described to the level of detail needed by the project team. These detailed descriptions provide the basis for the technical design because they explain how the actor and the system will behave to accomplish the goal of the use case.

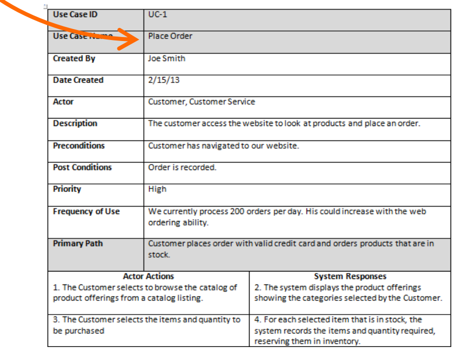
Defining and describing use cases is a progressively detailed process as outlined in the following steps.

1. Start with a use case on the diagram, outline the use case and identify the primary and alternate paths.
2. Begin to describe the detailed actions including how the actor will behave, how the system should respond, error messages and business rules.
3. Break the use cases into smaller use cases if needed. Some of the smaller use cases will be reusable by multiple other use cases; e.g. a login use case.

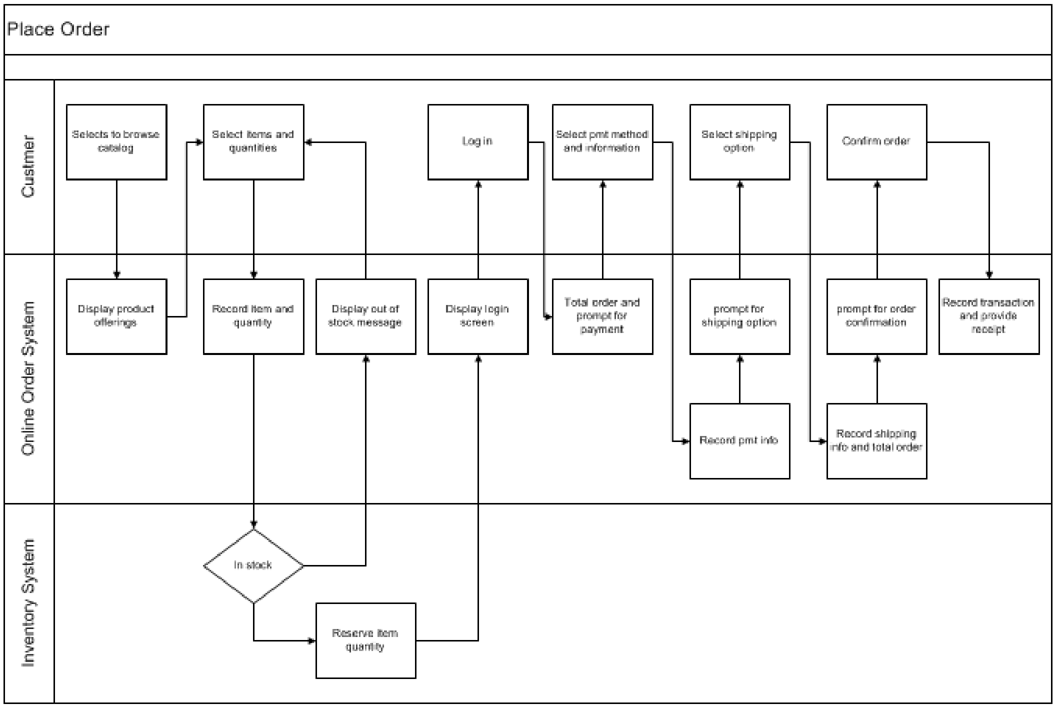
Use Case Diagram



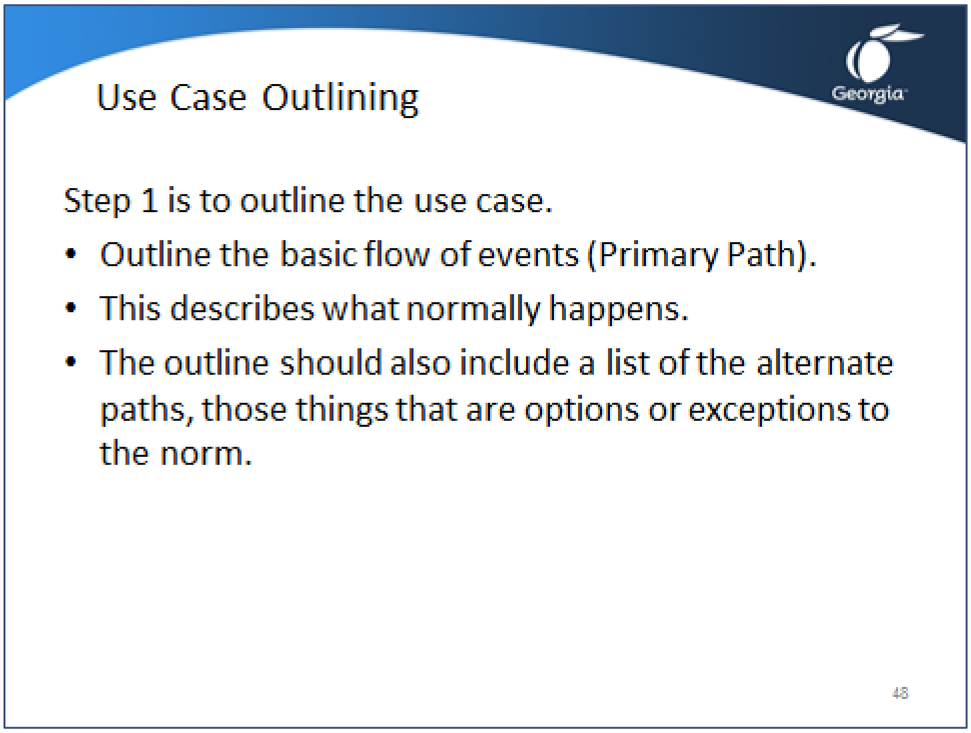
Use Case Description



Use Case ID	UC-1
Use Case Name	Place Order
Created By	Joe Smith
Date Created	2/15/13
Actor	Customer, Customer Service
Description	The customer access the website to look at products and place an order.
Preconditions	Customer has navigated to our website.
Post Conditions	Order is recorded.
Priority	High
Frequency of Use	We currently process 200 orders per day. His could increase with the web ordering ability.
Primary Path	Customer places order with valid credit card and orders products that are in stock.
Actor Actions	**System Responses**
1. The Customer selects to browse the catalog of product offerings from a catalog listing.	2. The system displays the product offerings showing the categories selected by the Customer.
3. The Customer selects the items and quantity to be purchased	4. For each selected item that is in stock, the system records the items and quantity required, reserving them in inventory.


Place Order
Custmer
Online Order System
Inventory System
Selects to browse catalog
Select items and quantities
Log in
Select pmt method and information
Select shipping option
Confirm order
Display product offerings
Record item and quantity
Display out of stock message
Display login screen
Total order and prompt for payment
prompt for shipping option
prompt for order confirmation
Record transaction and provide receipt
Record pmt info
Record shipping info and total order
In stock
Reserve item quantity

Topic 3: Developing Functional Requirements – Step 1 Use Case Outlining



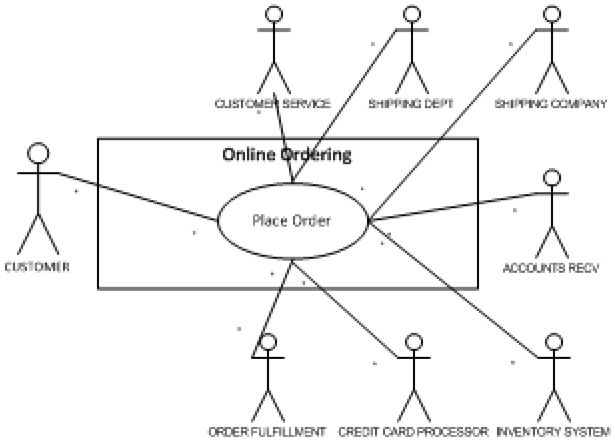
Use Case Outlining

Step 1 is to outline the use case.

- Outline the basic flow of events (Primary Path).
- This describes what normally happens.
- The outline should also include a list of the alternate paths, those things that are options or exceptions to the norm.

48

Step 1: Start with a use case on the diagram, outline the use case and identify primary and alternate paths. A work flow diagram may be used as an alternate to the outline.



Every use case has at least one path, the Primary Path. The Primary Path is the basic flow of events that describes what normally will happen when the use case is performed.

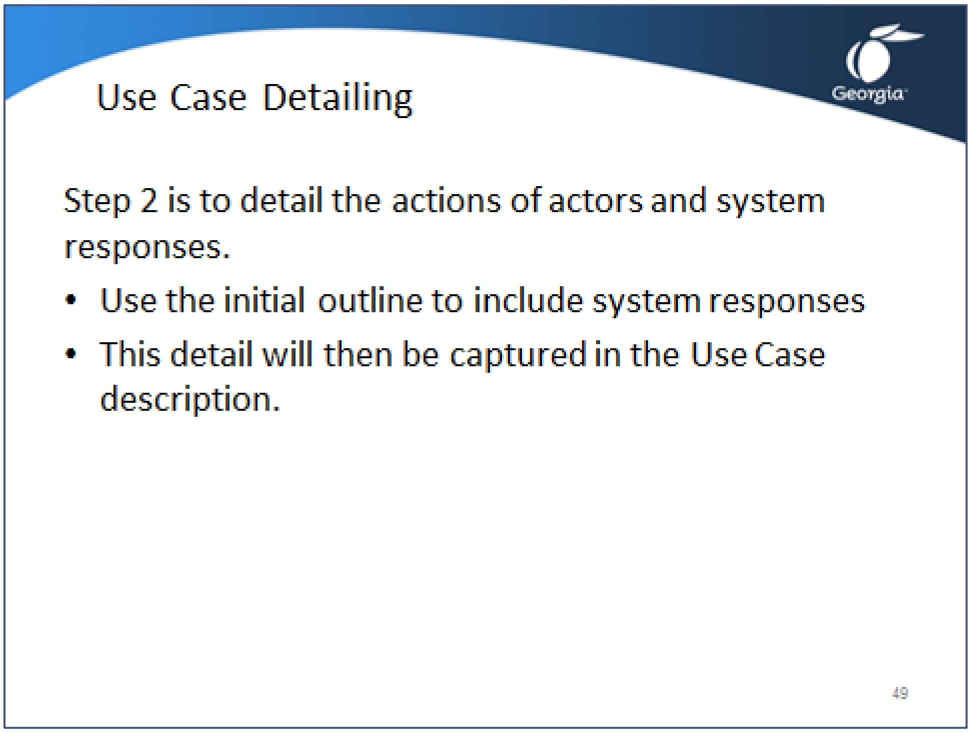
Example:

Use Case	UC-1: Place Order
Description:	An order is placed for office supplies via the website.
Outline Primary Path	
1. Customer browses product listings	
2. Customer selects one or more products to purchase	
3. Customer provides profile, shipping and payment information	
4. Customer confirms order	
5. Order is sent to order fulfillment	
6. Order is fulfilled and shipped	

The outline also includes a list of alternate paths and a brief description of each. An alternate path covers behavior that is optional or exceptional, is always dependent on some condition occurring in another flow of events, and shows something that interrupts the normal flow of events. To find the alternate paths ask “Is there another action that can be taken at this point?” or “Is there something that could go wrong at this point?”

Use Case	Place Order
Alternate Path	
A1	Invalid Credit Card
A2	Product not in stock
A3	Create customer profile
A4	Product on back order
A5	Confirmation fails
A6	Customer service representative places order on behalf of customer

Topic 3: Developing Functional Requirements – Step 2 Use Case Detailing



Use Case Detailing

Step 2 is to detail the actions of actors and system responses.

- Use the initial outline to include system responses
- This detail will then be captured in the Use Case description.

49

Step 2: Detail the actions of actors and system responses, include error messages and applicable business rules.

The initial outline will be used to fully detail the complete use case. Details will be added to both the primary flow and the alternate flow.

This detail is captured in the use case description template. A complete use case description can be complicated and may evolve over time. A sample template is displayed on the following pages.

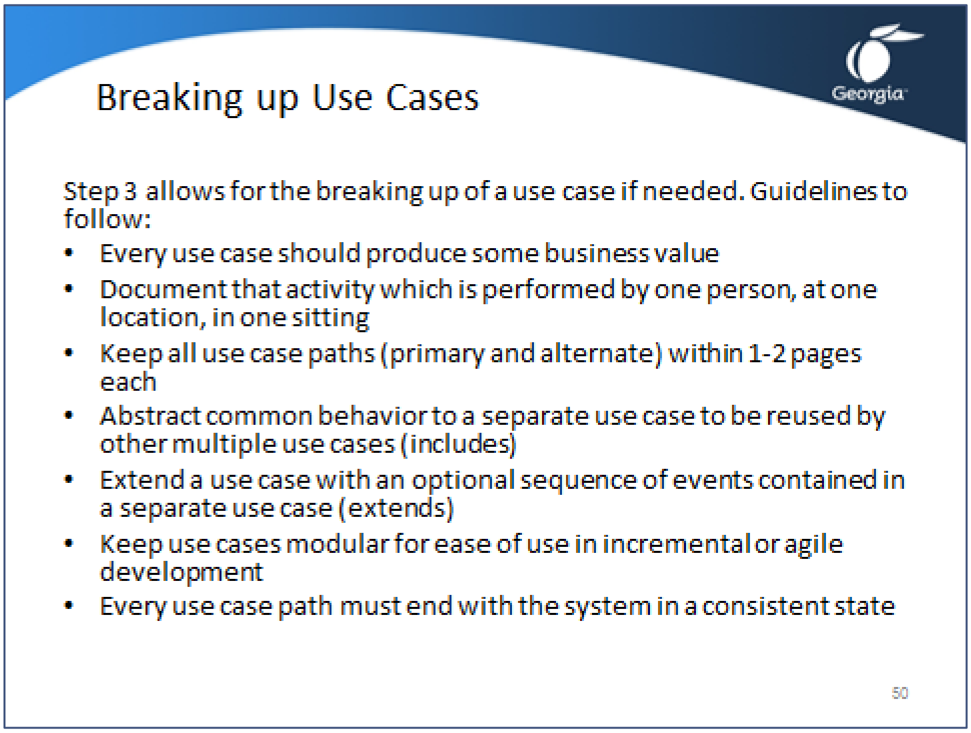
You should not expect to complete the use case description in just one meeting or interview. It may require several revisions.

USE CASE DESCRIPTION TEMPLATE

Use Case ID	A unique identifier that may be used to trace use cases to business requirements and test cases.
Use Case Name	A combination of and action verb and a noun phrase, written from the perspective of the actor, which describes the reason for the actor's interaction with the system.
Created By	Author of the use case.
Date Created	When the use case was created.
Actor	The primary actor initiates the use case. Actors that are supporting the use case or are offstage actors can be documented here as secondary actors.
Description	1 – 2 sentences used to concisely describe the goal of the primary actor.
Preconditions	The state of the system at the start of the use case. This is not an event or a trigger, but what must be true for the use case to be initiated.
Post Conditions	The state of the system at the end of the use case. This state must be verifiable.
Priority	The relative importance of this use case to the project. This alerts development to what is the most critical.
Frequency of Use	How often this system interaction occurs.
Primary Path	This describes a sequential flow of normal events depicting a dialog between the actor and the system.
Actor Actions 1. Describes the actor actions on the interface.	**System Responses** 2. Describes the system's responses to the actor's actions.
3.	4.
Alternate Path 1	Describes a sequential flow of actions related to a specific condition, under which the system exhibits a different behavior.
Actor Actions 1. Note where these steps are the same as or different from the primary path.	**System Responses** 2. Note where these steps are the same as or different from the primary path.
3.	4.
Post Conditions	Indicate if the alternate path has ended with a different result.
Alternate Path 2	Describes a sequential flow of actions related to a specific condition, under

<table>
<tr><td></td><td colspan="3">which the system exhibits a different behavior.</td></tr>
<tr><td colspan="2"><b>Actor Actions</b>
1. Note where these steps are the same as or different from the primary path.</td><td colspan="2"><b>System Responses</b>
2. Note where these steps are the same as or different from the primary path.</td></tr>
<tr><td colspan="2">3.</td><td colspan="2">4.</td></tr>
<tr><td><b>Post Conditions</b></td><td colspan="3">Indicate if the alternate path has ended with a different result.</td></tr>
<tr><td><b>Additional Notes</b></td><td colspan="3">Any information that the technical team should know about the use case.</td></tr>
<tr><td><b>Revision History</b></td><td colspan="2"><b>Revised by</b></td><td><b>Date</b></td></tr>
<tr><td></td><td colspan="2">Include the person who made the change.</td><td>Include the date of the change.</td></tr>
</table>

Topic 3: Developing Functional Requirements – Step 3 Breaking up Use Cases



Breaking up Use Cases

Step 3 allows for the breaking up of a use case if needed. Guidelines to follow:

- Every use case should produce some business value
- Document that activity which is performed by one person, at one location, in one sitting
- Keep all use case paths (primary and alternate) within 1-2 pages each
- Abstract common behavior to a separate use case to be reused by other multiple use cases (includes)
- Extend a use case with an optional sequence of events contained in a separate use case (extends)
- Keep use cases modular for ease of use in incremental or agile development
- Every use case path must end with the system in a consistent state

50

Step 3: Break up the use case into smaller use cases if needed.

Some use cases can become very large and to improve readability and understanding it is best to decide how to simplify the base use case. To do this the following guidelines should be followed.

- Every use case should produce some business value
- Document that activity which is performed by one person, at one location, in one sitting
- Keep all use case paths (primary and alternate) within 1-2 pages each
- Abstract common behavior to a separate use case to be reused by other multiple use cases (includes)
- Extend a use case with an optional sequence of events contained in a separate use case (extends)
- Keep use cases modular for ease of use in incremental or agile development
- Every use case path must end with the system in a consistent state

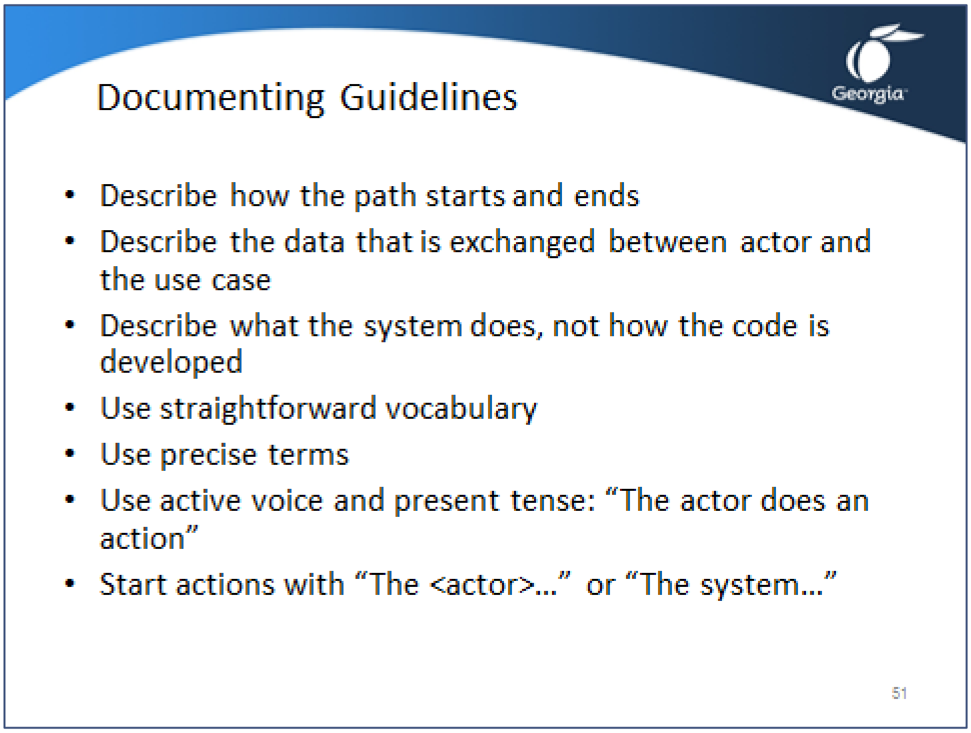
INCLUDES USE CASE

Use cases that are used by other use cases are called *includes*. The *includes* use case is used to document common, reusable functions among other use case. An *includes* use case can live on its own.

EXTENDS USE CASE

An *extends* use case is additional functionality the base use case can execute. An *extends* use case relates to only one base use case.

Topic 3: Developing Functional Requirements – Documenting Guidelines



Documenting Guidelines

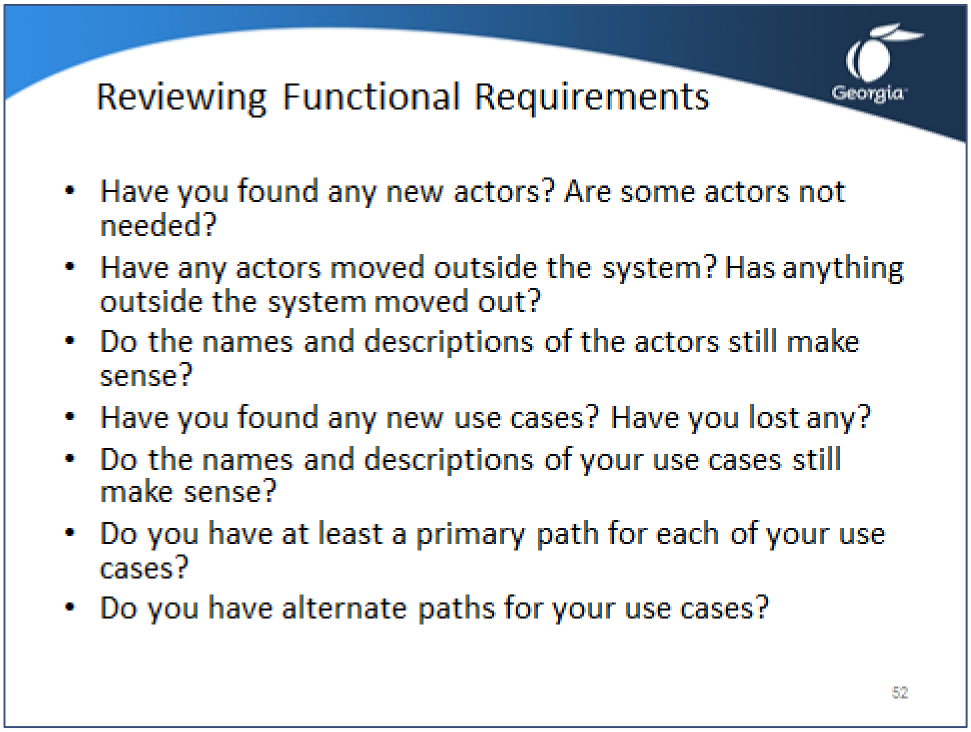
- Describe how the path starts and ends
- Describe the data that is exchanged between actor and the use case
- Describe what the system does, not how the code is developed
- Use straightforward vocabulary
- Use precise terms
- Use active voice and present tense: "The actor does an action"
- Start actions with "The <actor>…" or "The system…"

51

Use cases are complete only when the stakeholders can agree that their needs are met by the use case, the developers can agree that they can build a system from the use case description, and testers can test the use case description. Think of the use case as a conversation.

- Describe how the path starts and ends
- Describe the data that is exchanged between actor and the use case
- Describe what the system does, not how the code is developed
- Use straightforward vocabulary
- Use precise terms
- Use active voice and present tense: "The actor does an action"
- Start actions with "The <actor>…" or "The system…"

Topic 3: Developing Functional Requirements – Reviewing Functional Requirements



Reviewing Functional Requirements

- Have you found any new actors? Are some actors not needed?
- Have any actors moved outside the system? Has anything outside the system moved out?
- Do the names and descriptions of the actors still make sense?
- Have you found any new use cases? Have you lost any?
- Do the names and descriptions of your use cases still make sense?
- Do you have at least a primary path for each of your use cases?
- Do you have alternate paths for your use cases?

52

Questions to ask during a review of functional requirements

- Have you found any new actors? Are some actors not needed?
- Have any actors moved outside the system? Has anything outside the system moved out?
- Do the names and descriptions of the actors still make sense?
- Have you found any new use cases? Have you lost any?
- Do the names and descriptions of your use cases still make sense?
- Do you have at least a primary path for each of your use cases?
- Do you have alternate paths for your use cases?

Questions to ask during a review of use cases

- Is the name of the goal of the primary actor an active verb/noun phrase?
- Can the system deliver that goal?
- Does the use case content match the stated goal level?
- Are preconditions mandatory, and can they be set in place by the system under design?
- Is it true that preconditions are never checked in the use case?
- Does the primary path have 3-9 steps?
    - Does it run from the starting event to “post condition” success?
- Have you considered all of the alternatives to the primary path that are possible?
- All steps in all paths:
    - Is it phrased as a goal that succeeds?
    - Does the process move distinctly forward after its successful completion?
    - Is it clear which actor is operating the goal?
    - Is the intent of the actor clear?
- Can and must the system both detect and handle an extension condition?
- About the content
    - To the sponsors and users: “Is this what you want?”

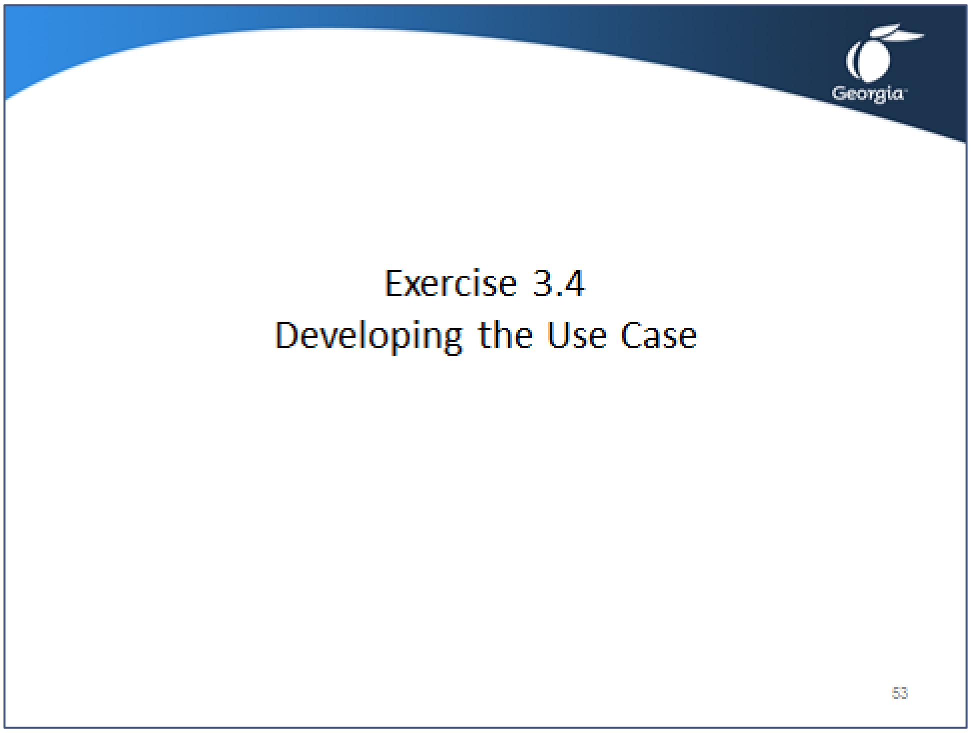
  - To the sponsors and users: “Will you be able to tell, upon delivery, whether you got this?”
  - To the developers: “Can you implement this?”

SUMMARY OF USE CASE DEVELOPMENT

Use cases describe observable behaviors of the solution. Their development includes the following:

1. Confirm actors and goals
2. Develop an outline of the use case
3. Write a brief description of the use case
4. Record preconditions and post-conditions
5. Detail the basic flow
6. Detail the alternate flows
7. Review the use case
8. Break down large use cases when appropriate

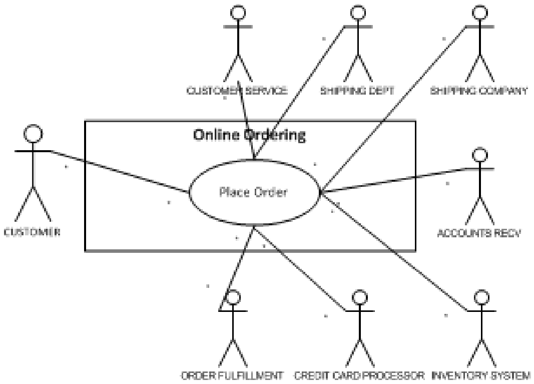
Exercise 3.4 Developing the Use Case



Exercise 3.4
Developing the Use Case

53

Using the information provided in the Case Study, the Use Case Description template below, and Exercises 3.1 and 3.2 develop a use case for the Place an Order process. Also, update the Requirements Traceability Matrix.



Use Case Template

<table>
<tr><td><b>Use Case ID</b></td><td colspan="3"></td></tr>
<tr><td><b>Use Case Name</b></td><td colspan="3"></td></tr>
<tr><td></td><td colspan="3"></td></tr>
<tr><td><b>Primary Path</b></td><td colspan="3"></td></tr>
<tr><td colspan="2"><b>Actor Actions</b></td><td colspan="2"><b>System Responses</b></td></tr>
<tr><td colspan="2"></td><td colspan="2"></td></tr>
<tr><td colspan="2"></td><td colspan="2"></td></tr>
<tr><td colspan="2"></td><td colspan="2"></td></tr>
<tr><td colspan="2"></td><td colspan="2"></td></tr>
<tr><td><b>Alternate Path 1</b></td><td colspan="3">Describes a sequential flow of actions related to a specific condition, under which the system exhibits a different behavior.</td></tr>
<tr><td colspan="2"><b>Actor Actions</b></td><td colspan="2"><b>System Responses</b></td></tr>
<tr><td colspan="2"></td><td colspan="2"></td></tr>
<tr><td colspan="2"></td><td colspan="2"></td></tr>
<tr><td><b>Post Conditions</b></td><td colspan="3"></td></tr>
<tr><td><b>Additional Notes</b></td><td colspan="3"></td></tr>
<tr><td><b>Revision History</b></td><td colspan="2"><b>Revised by</b></td><td><b>Date</b></td></tr>
<tr><td></td><td colspan="2"></td><td></td></tr>
</table>

Requirements Traceability Matrix

Requirements Traceability Matrix	
Project Name:	Speedy Office Supplies Web Expansion Project
Project Description:	Create the retail shopping experience for customers on our website.
Cost Center:	Customer Service Department

ID	Requirements Description	Use Case ID	Use Case Name	Test Case ID	Test Case Name	Date Tested	Approved By
R1							
R2							
R3							
R4							
R5							
R6							
R7							

Lesson 3 Summary: Learning Objectives Recap

- **Recognize the core components of requirements**
  There are several core components of requirements as described below:
  - DATA (Entities and Attributes)
    - An **entity** is a uniquely identifiable person, thing, or concept whose information is important to the business.
    - An **attribute** is a piece of information that is a characteristic of an entity.
  - PROCESSES or USE CASES
    - A **process** is a business activity that transforms inputs (entities and attributes) into outputs.
    - A **Use Case** is a high level process that will be included in the software automation.
  - EXTERNAL AGENTS or ACTORS
    - An **external agent** is a person, organization, or system with which the business area interacts.
    - An **actor** is any resource that uses software automation.
  - BUSINESS RULES
    - A **business rule** is a condition that governs the way work is done

- **Explain the components and purpose of Product Scope**
  Product scope as defined by the *PMBOK® Guide* is "the features and functions that characterize a product, service, or result." Determining the product scope involves prioritization and consensus. The team may use the following steps iteratively as needed to move from business requests to the initial solution plan
  1. Review/discuss each requirement request to ensure the business needs are clearly understood in order to estimate effectively.
  2. Discuss the business priority for each item with the domain SMEs
  3. Ask the IT team for technical priorities and estimated costs/time for each item
  4. Assist the team in breaking the list into delivery approaches, such as releases or iterations, based on priorities and budget constraints
  5. Gain agreement from all stakeholders.

- **Describe how to develop and communicate Functional Requirements**
  Functional Requirements describe the behavior and information that the solution will manage.
  The Use Case Model is a technique for defining and documenting functional requirements.
  A Use Case describes the *interaction between an actor and a system* that accomplishes a business goal from the actor's perspective.

Notes

LESSON 4: NON-FUNCTIONAL AND TRANSITIONAL REQUIREMENTS

Topic 1: Developing Non-Functional Requirements

Topic 2: Developing Transitional Requirements

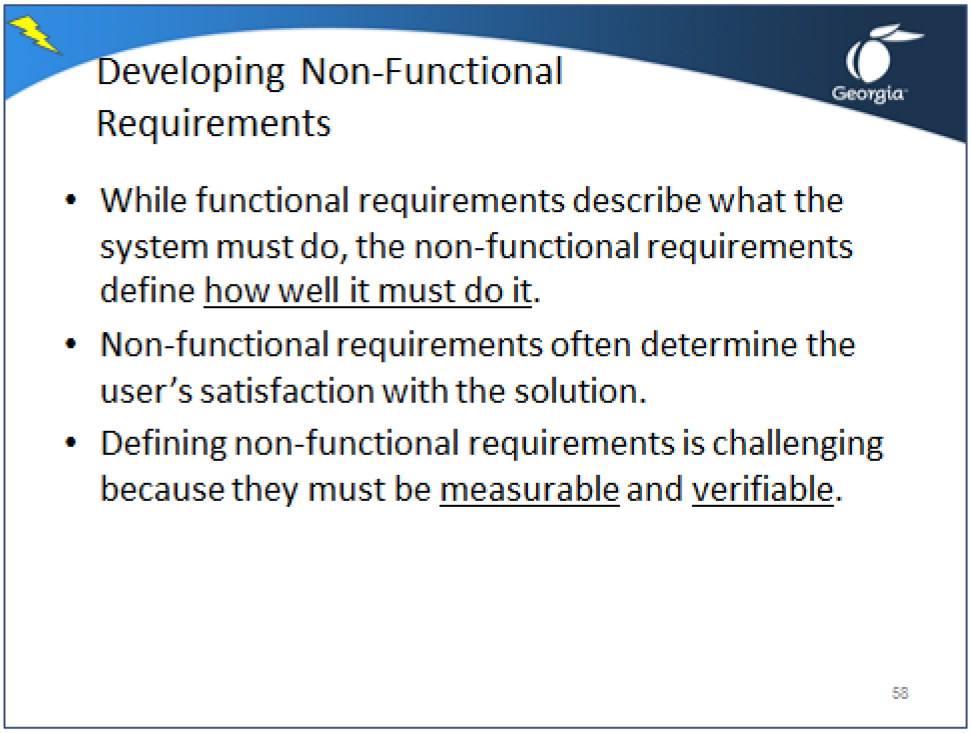
Student Learning Objectives

After completing this lesson you should be able to

- Describe how to develop and communicate Non-functional requirements
- Describe how to develop and communicate Transitional requirements

Approximate Presentation time: 2 hours

Topic 1: Developing Non-Functional Requirements



Developing Non-Functional Requirements

- While functional requirements describe what the system must do, the non-functional requirements define how well it must do it.
- Non-functional requirements often determine the user's satisfaction with the solution.
- Defining non-functional requirements is challenging because they must be measurable and verifiable.

58

While functional requirements describe what the system must do, the non-functional requirements define how well it must do it.

Non-functional requirements often determine the user's satisfaction with the solution. Defining non-functional requirements is challenging because they must be measurable and verifiable.

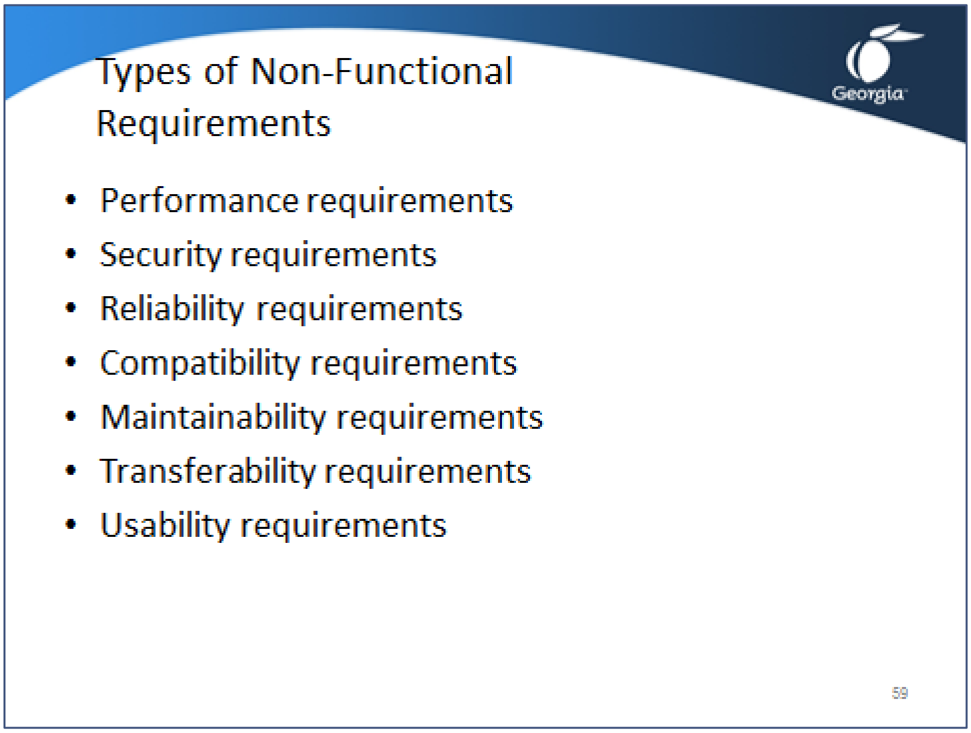
Non-functional requirements state various system and development constraints that are not addressed by business, functional, or technical requirements.

These requirements can apply to a system on several levels:

1. The entire system
   a. These can be quality, cost, reliability, performance, or security goals.
2. Single or grouped use cases
   a. These are additional qualities that relate to specific use cases, e.g. time constraints, security access, or performance (throughput, peak traffic).

The project manager needs to ensure non-functional requirements are not overlooked or assumed.

Topic 1: Developing Non-Functional Requirements (continued)



Types of Non-Functional Requirements

- Performance requirements
- Security requirements
- Reliability requirements
- Compatibility requirements
- Maintainability requirements
- Transferability requirements
- Usability requirements

59

Below are some non-functional requirements that are important to discuss:

- Performance requirements
- Security requirements
- Reliability requirements
- Compatibility requirements
- Maintainability requirements
- Transferability requirements
- Usability requirements

Performance Requirements

Performance requirements describe how well the solution should *perform*.
Example:

Reqmt number	Requirement	Category/type
29	The number of simultaneous users will be less than 2000.	Performance
30	95% of database queries shall be completed within 2 seconds.	Performance
31	The system shall be able to process 100 payment transactions per second in peak load.	Performance

Security Requirements

Security requirements typically fall within the areas of authorization, privacy/confidentiality, user authentication, and integrity of information.
Example:

Reqmt number	Requirement	Category/type
32	Only the Training Administrator may add, change, or delete records from the Student and Course profiles.	Security
33	Only a valid student may submit a class for registration.	Security
34	The length of a student password must be between 8 and 15 characters and consist of at least one upper case character, one symbol and one number.	Security

Questions to ask:

1. Who can add, change or delete data elements?
2. Who can utilize each function?
3. What conditions must be met to allow access?
4. What tracking should be formed?
5. Who will maintain the security access?
6. How many levels of access are needed?
7. What will be reported as a breach?

Reliability Requirements

Reliability requirements describe the needed availability of the solution.

- When is the system expected to be available?
- Ability to recover errors
- Percentage of time available (e.g. 99%)
- Downtime windows for maintenance and upgrades

Example:

Reqmt number	Requirement	Category/type
45	The training registration site shall be available 99% of the time.	Reliability
46	The online payment request system shall be available from 6:00 am EST to 11:00 pm EST.	Reliability

Measurements:

1. Time periods (i.e. days, hours)
2. Errors (percentage, counts)

Questions to ask:

1. What is the impact on the business when the system is down?
2. What housekeeping tasks will be done on a regular basis (e.g. backups)? Can they be done while the system is operating?
3. When is the best time to perform maintenance?
4. What notification is needed when the system is down?
5. Who should receive notification?
6. How much advance warning do users need for a planned outage?
7. How quickly will the operations team respond to an unexpected outage?

Compatibility Requirements

Compatibility refers to the extent to which the solution will coexist with other systems.

- Use of common standards, common technology, protocols, etc.
- Ability to work with other systems regardless of their programming language or hardware platform
- Ability to exchange data

Example:

Reqmt number	Requirement	Category/type
55	The system must be able to interface with any HTML browser.	Compatibility
56	The website must operate correctly with the most recent and past 2 revisions of IE.	Compatibility

Measurements:

1. Common protocols (yes/no)
2. Compatible hardware, operating systems, browsers, etc.

Questions to ask:

1. How would other departments or people use the data from this system?
2. What kinds of data exchange are envisioned?
3. What information must be exchanged with other systems?
4. Which external suppliers or customers use interfaces?
5. Who will build each interface?
6. What agreements (contracts, Service Level Agreements (SLA)) have been put into place?

MAINTAINABILITY REQUIREMENTS

Maintainability refers to how easy it is to operate and repair the system.

- Ease of testing and de-bugging
- Ease of future modifications
- Ability to change one component without affecting others
- Re-usability of components
- Ability to expand or upgrade capabilities

Example:

Reqmt number	Requirement	Category/type
67	The development process must have a regression test procedure which allows complete re-testing within two business days.	Maintainability
68	A new policy type code must be able to be added to the system within 3 business days.	Maintainability

Measurements:

1. Number of reusable modules
2. Testing checkpoints included
3. Time to repair
4. Frequency of maintenance

Questions to ask:

1. What maintenance records should be kept?
2. What effect do maintenance activities have on customers?
3. Who performs system upgrades?
4. Who will migrate enhancement releases?
5. Who is responsible for the interfaces?
6. How often will new releases be developed?
7. How quickly will vendor upgrades be installed?
8. Which features are most likely to change?

TRANSFERABILITY REQUIREMENTS

Transferability is the ease with which a system can be transferred to a different hardware or software environment.

- Can the system be installed in another environment?
- Can the system be installed at a different location/different geography?

Example:

Reqmt number	Requirement	Category/type
77	The product is targeted for sale in the European market next year.	Transferability
78	The system shall be developed for MS Windows® and Microsoft® operating system platforms.	Transferability

Measurements:

1. Specific list of environments or platforms
2. Specific list of sites or locations

Questions to ask:

1. Who will be porting the system to different environments or locations?
2. What will be different when moving to a new platform?
3. What operating environment is considered the base environment?
4. Will data files be transferable?
5. Will the code run exactly the same way on all platforms?
6. Will the system have the same look and feel across platforms?
7. Will the same developers maintain all platforms?
8. What government regulations need to be addressed?

USABILITY REQUIREMENTS

Usability is the ease with which the user is able to learn, operate, and interpret results from the system.

- Ease of entry
- Ease of learning
- Ease of handling
- Intuitiveness

Example:

Reqmt number	Requirement	Category/type
87	A novice user must be able to add new employee to the payroll system within 10 minutes.	Usability
88	An experienced user must be able to update an employee's withholding parameters within 3 minutes.	Usability

Measurements:

- Time to get over learning curve
- Problem counts
- Task time
- Keystroke counts

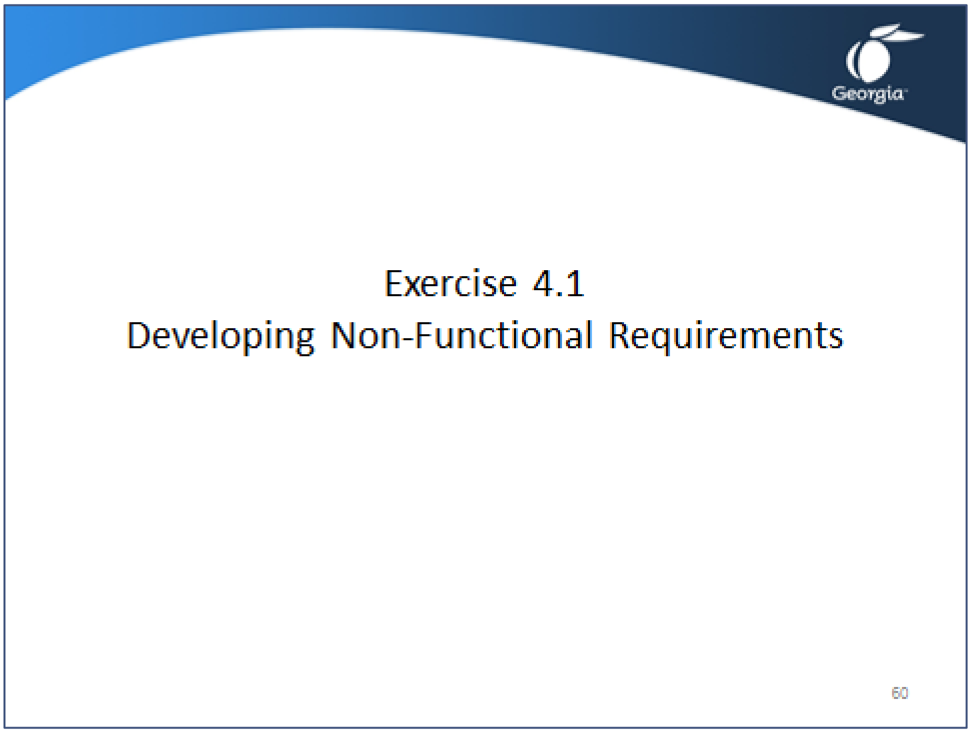
COTS NON-FUNCTIONAL REQUIREMENTS

Defining other non-functional requirements is very important when selecting a vendor application. Non-functional requirements should be included in the vendor request or RFP (Request for Proposal).

- Flexibility – ability to tailor/customize (configure)
  - What system settings can be added at the user level?
  - What types of new users might be added?
- Maintainability
  - Who is responsible for monitoring and correcting faults?
  - Who is responsible for upgrades?
- Scalability – can the system expand processing capabilities to support business growth?

- Expanding business locations
- Increased number of users
- Ability to add hardware

Exercise 4.1 Develop Non-Functional Requirements



Exercise 4.1

Developing Non-Functional Requirements

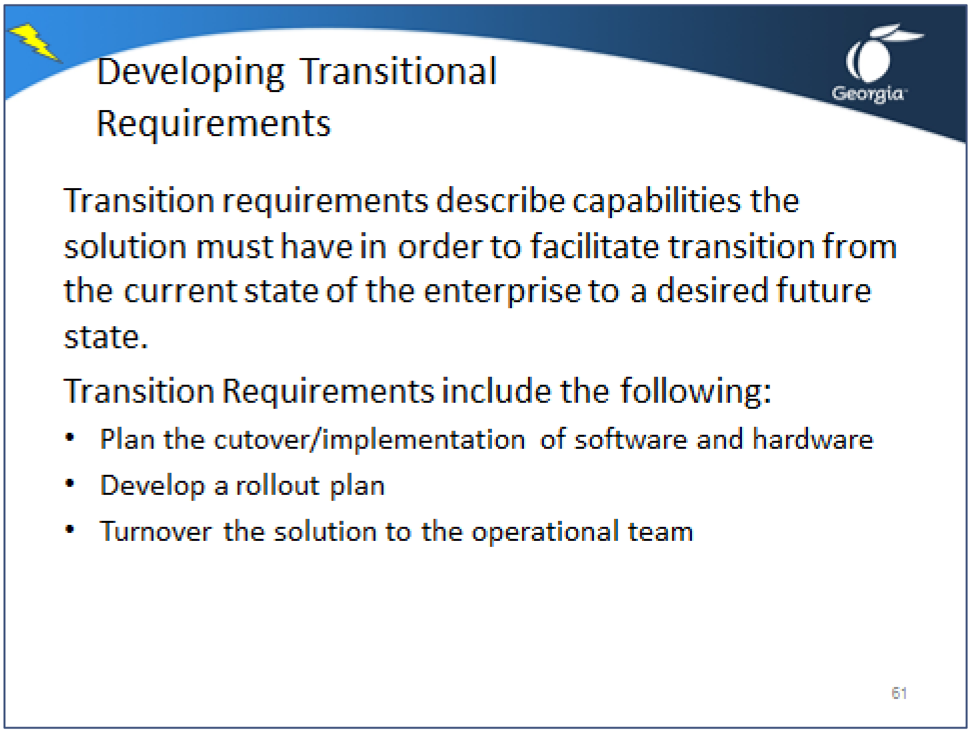
60

Using the information provided in the Case Study and the template provided on the next page, develop the non-functional requirements.

Non-Functional Requirements

Reqmt number	Requirement	Category/type

Topic 2: Developing Transitional Requirements



Developing Transitional Requirements

Transition requirements describe capabilities the solution must have in order to facilitate transition from the current state of the enterprise to a desired future state.

Transition Requirements include the following:

- Plan the cutover/implementation of software and hardware
- Develop a rollout plan
- Turnover the solution to the operational team

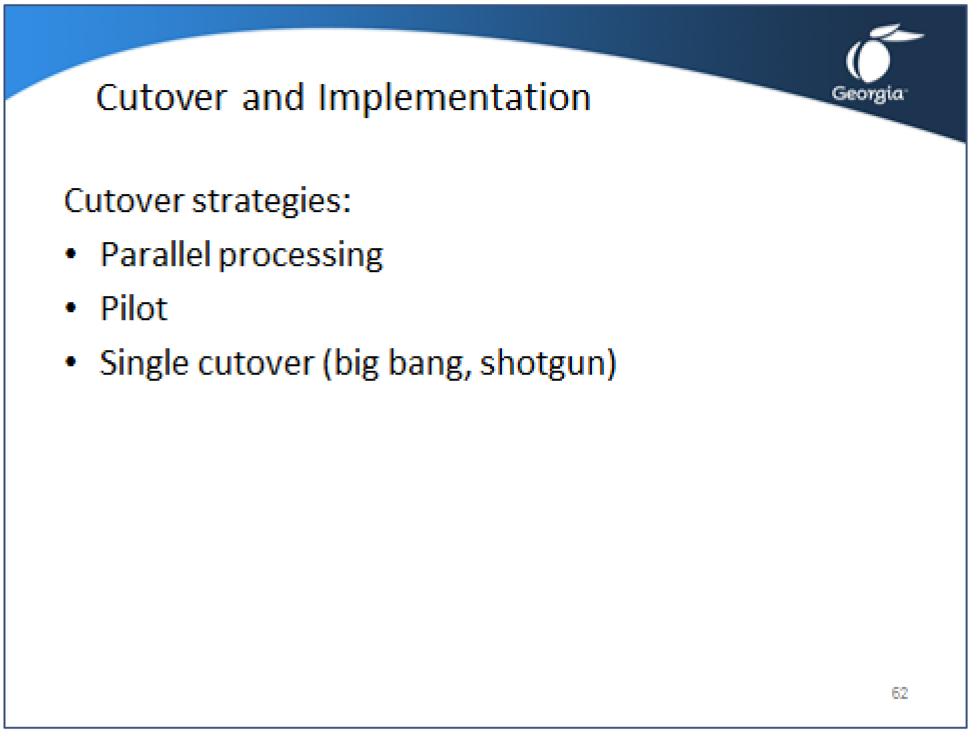
“Transition requirements describe capabilities the solution must have in order to facilitate transition from the current state of the enterprise to a desired future state.” *Guide to the BABOK®*

- Transition requirements are temporary – they will not be needed after the transition is made
- They are sometimes the last requirements developed because they are dependent on an understanding of the current and future state
- They include things like data conversion requirements, software installation and rollover plans, and user training

Transition Requirements include the following:

- Plan the cutover/implementation of software and hardware
- Develop a rollout plan
- Turnover the solution to the operational team

Topic 2: Developing Transitional Requirements – Cutover/Implementation



Cutover and Implementation

Cutover strategies:

- Parallel processing
- Pilot
- Single cutover (big bang, shotgun)

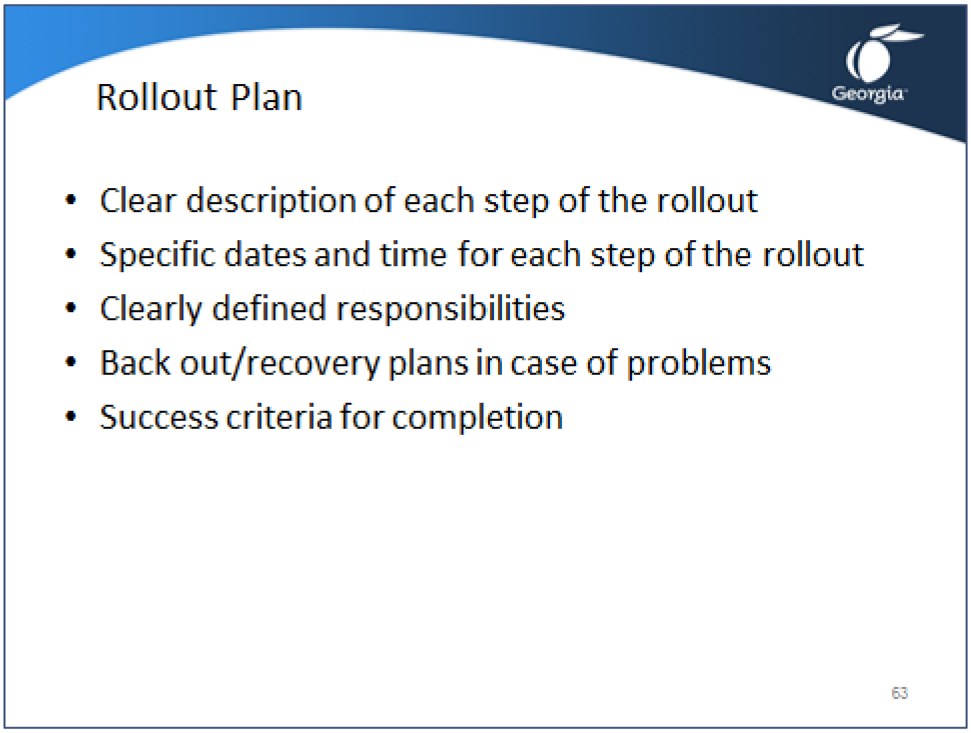
62

When installing new software there are many approaches. The team should discuss the options and agree on the best approach for the project.

Cutover strategies:

- Parallel processing
- Pilot
- Single cutover (big bang, shotgun)

Topic 2: Developing Transitional Requirements – Rollout Plan



Rollout Plan

- Clear description of each step of the rollout
- Specific dates and time for each step of the rollout
- Clearly defined responsibilities
- Back out/recovery plans in case of problems
- Success criteria for completion

63

“The focus of Release Management is the protection of the live environment and its services through the use of formal procedures and checks.” ITIL Service Support Manual

Include:

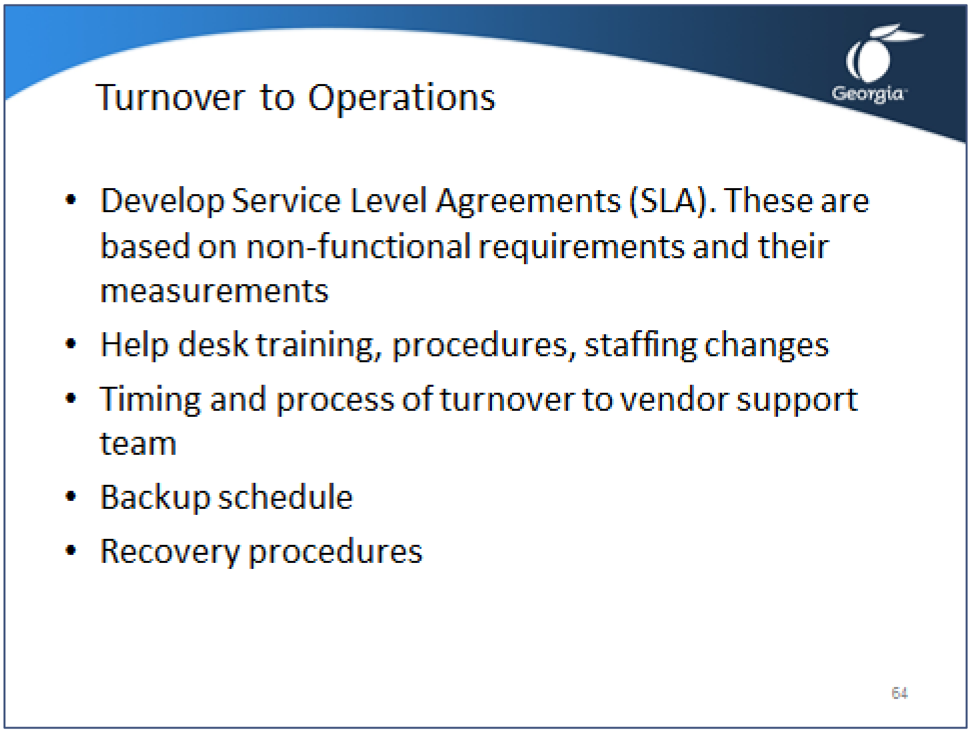
+ Clear description of each step of the rollout
+ Specific dates and time for each step of the rollout
+ Clearly defined responsibilities
+ Back out/recovery plans in case of problems
+ Success criteria for completion

Include operations personnel, business sponsor and users in the planning.

Sample Rollout Plan

Date/Time	Task	Responsible Person	Back-out/contingency
April 3 – Friday at noon	All users log out of the system after finishing the weeks data entry work.	John – Business area representative	If work cannot be completed, keep paper backup of information so that it can be entered into the new system after the cutover.
April 3 – Friday 12:30 – 1:30 pm	Backup database. Print pre-conversion counts.	Jeff – IT DBA	Backup has been tested and timed at about 45 minutes.
April 3 – Friday 1:30 – 5 pm	Convert production data to new database format.	Donna – IT Dev	Conversion programs have been tested and timed at about 2 hours.
	Review error report from conversion and make corrections.	Donna – IT Dev John – Business	Data cleanup should take no longer than 1 hour.
	Finalize conversion. Print post conversion counts.	Donna – IT Dev	Pre-conversion counts should match post-conversion counts. If not – find discrepancy and correct if possible. If more than 10% errors, rollout will be cancelled and rescheduled for future.

Topic 2: Developing Transitional Requirements – Turnover to Operations



Turnover to Operations

- Develop Service Level Agreements (SLA). These are based on non-functional requirements and their measurements
- Help desk training, procedures, staffing changes
- Timing and process of turnover to vendor support team
- Backup schedule
- Recovery procedures

64

"Transfer the project's products, services, or results…to production and/or operations." *Guide to the PMBOK®*

The change management plan must include turnover to the operational support group. This turnover may include:

+ Develop Service Level Agreements (SLA). These are based on non-functional requirements and their measurements

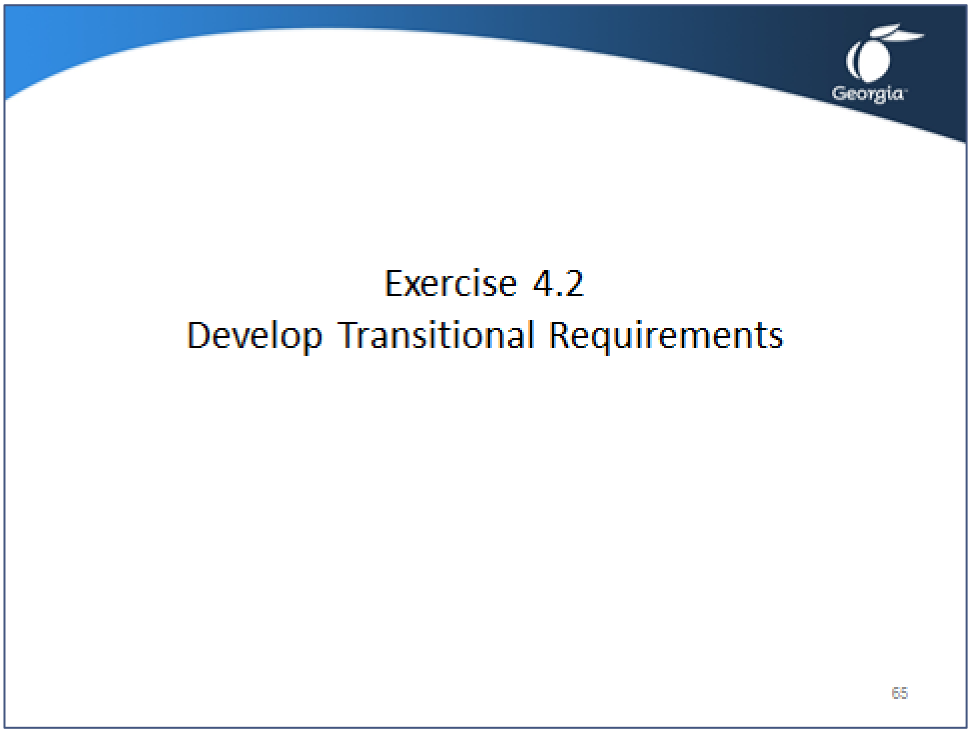
  Example:

  Non-functional requirement – The number of simultaneous users will be less than 2000.

  SLA – When the number of simultaneous users is less than 2000, agreed upon login times must be met. If they are not met or users under 2000 are not able to log in, the help desk will be notified and the problem will be addressed within one business day.

+ Help desk training, procedures, staffing changes
+ Timing and process of turnover to vendor support team
+ Backup schedule
+ Recovery procedures

Exercise 4.2 Develop Transitional Requirements



Exercise 4.2
Develop Transitional Requirements

65

Determine the transitional requirements for the project case study.

- Plan the cutover/implementation of software and hardware
- Develop a rollout plan
- Turnover the solution to the operational team

Lesson 4 Summary: Learning Objectives Recap

- **Describe how to develop and communicate Non-functional requirements**
  Non-functional requirements often determine the user's satisfaction with the solution. Defining non-functional requirements is challenging because they must be measurable and verifiable.

  These requirements can apply to a system on several levels:
  - The entire system
    - These can be quality, cost, reliability, performance, or security goals.
  - Single or grouped use cases
    - These are additional qualities that relate to specific use cases, e.g. time constraints, security access, or performance (throughput, peak traffic).

- **Describe how to develop and communicate Transitional Requirements**
  Transition requirements describe capabilities the solution must have in order to facilitate transition from the current state of the enterprise to a desired future state.

  Transition Requirements include the following:
  - Plan the cutover/implementation of software and hardware
  - Develop a rollout plan
  - Turnover the solution to the operational team

Notes

LESSON 5: VALIDATING PROJECT REQUIREMENTS

Topic 1: Testing Terminology

Topic 2: When Testing Occurs

Topic 3: Developing the Test Plan

Topic 4: Developing Test Cases and Scripts

Topic 5: Validating Scope using the Traceability Matrix

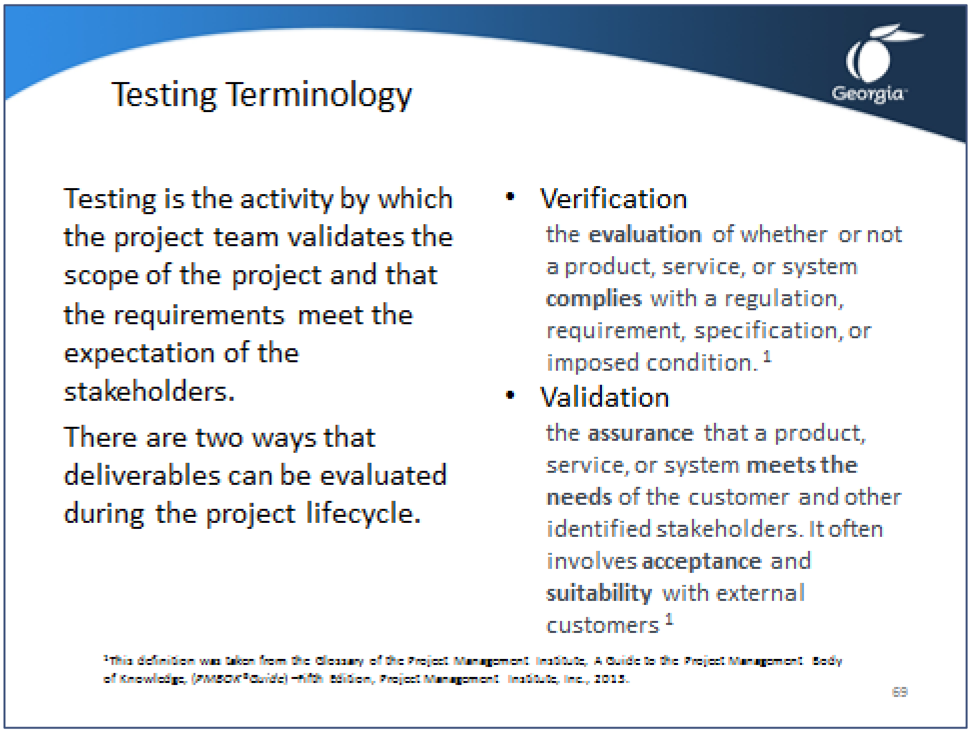
Student Learning Objectives

After completing this lesson you should be able to

- Discuss testing terminology
- Discuss various testing methods to validate requirements
- Recognize how to develop a test plan for gathered requirements
- Explain how to develop Test Cases and Scripts to validate requirements
- Explain how to validate requirements and scope using the traceability matrix

Approximate Presentation time: 2 hours

Topic 1: Testing Terminology



Testing Terminology

Testing is the activity by which the project team validates the scope of the project and that the requirements meet the expectation of the stakeholders.

There are two ways that deliverables can be evaluated during the project lifecycle.

- Verification
  the **evaluation** of whether or not a product, service, or system **complies** with a regulation, requirement, specification, or imposed condition.[1]
- Validation
  the **assurance** that a product, service, or system **meets the needs** of the customer and other identified stakeholders. It often involves **acceptance** and **suitability** with external customers[1]

[1] This definition was taken from the Glossary of the Project Management Institute, A Guide to the Project Management Body of Knowledge, (*PMBOK®Guide*) –Fifth Edition, Project Management Institute, Inc., 2013.

69

In order to validate the scope of a project (requirements) various tests should be performed to make sure the expectation of stakeholders are being met. This topic will discuss the testing terms you will need to understand and a testing lifecycle to use that will assure that project requirements have been met.

Verification
PMBOK® Guide states that verification is "the evaluation of whether or not a product, service, or system complies with a regulation, requirement, specification, or imposed condition. It is often an internal process."[1]

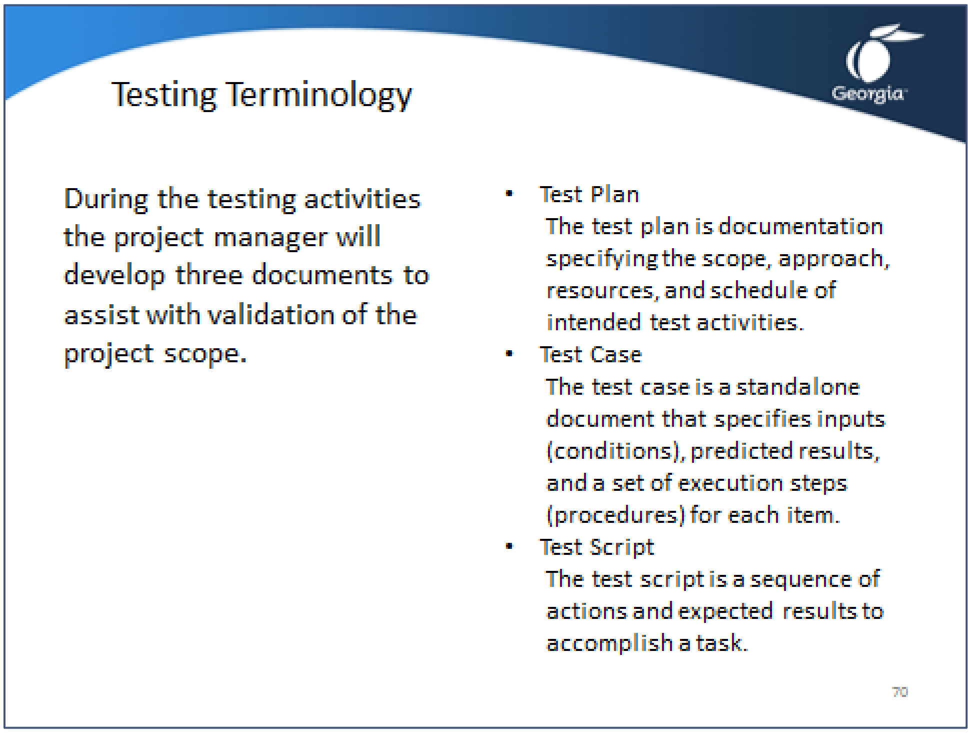
Verification is the process of inspecting deliverables throughout the lifecycle. Inspections and walkthroughs are examples of verification techniques. Verification is usually performed by inspecting without execution on the computer (static testing).

Validation
PMBOK® Guide states that validation is "the assurance that a product, service, or system meets the needs of the customer and other identified stakeholders. It often involves acceptance and suitability with external customers."[1]

Validation is the process of executing something to see how it behaves. Unit, integration, system, and acceptance testing are examples of validation techniques. Validation is usually performed by execution on a computer (dynamic testing).

[1] This definition was taken from the Glossary of the Project Management Institute, A Guide to the Project Management Body of Knowledge, (*PMBOK®Guide*) –Fifth Edition, Project Management Institute, Inc., 2013.

Topic 1: Testing Terminology …continued



Testing Terminology

During the testing activities the project manager will develop three documents to assist with validation of the project scope.

- Test Plan
  The test plan is documentation specifying the scope, approach, resources, and schedule of intended test activities.
- Test Case
  The test case is a standalone document that specifies inputs (conditions), predicted results, and a set of execution steps (procedures) for each item.
- Test Script
  The test script is a sequence of actions and expected results to accomplish a task.

70

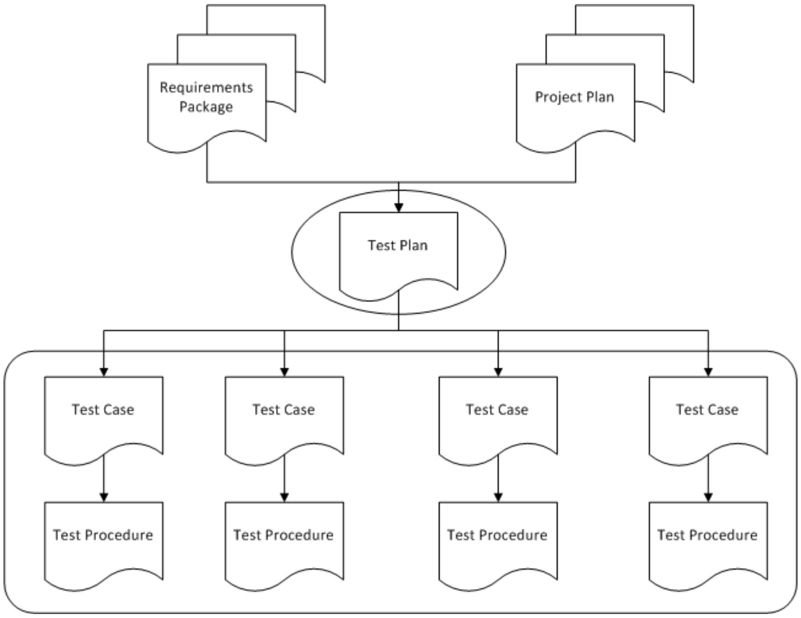
Test Plan
The test plan is documentation specifying the scope, approach, resources, and schedule of intended test activities. It defines test items, the features to be tested, the testing tasks, responsibilities, required resources, and any risks requiring contingency planning.

Test Case
The test case is a standalone document that specifies inputs (conditions), predicted results, and a set of execution steps (procedures) for each item.

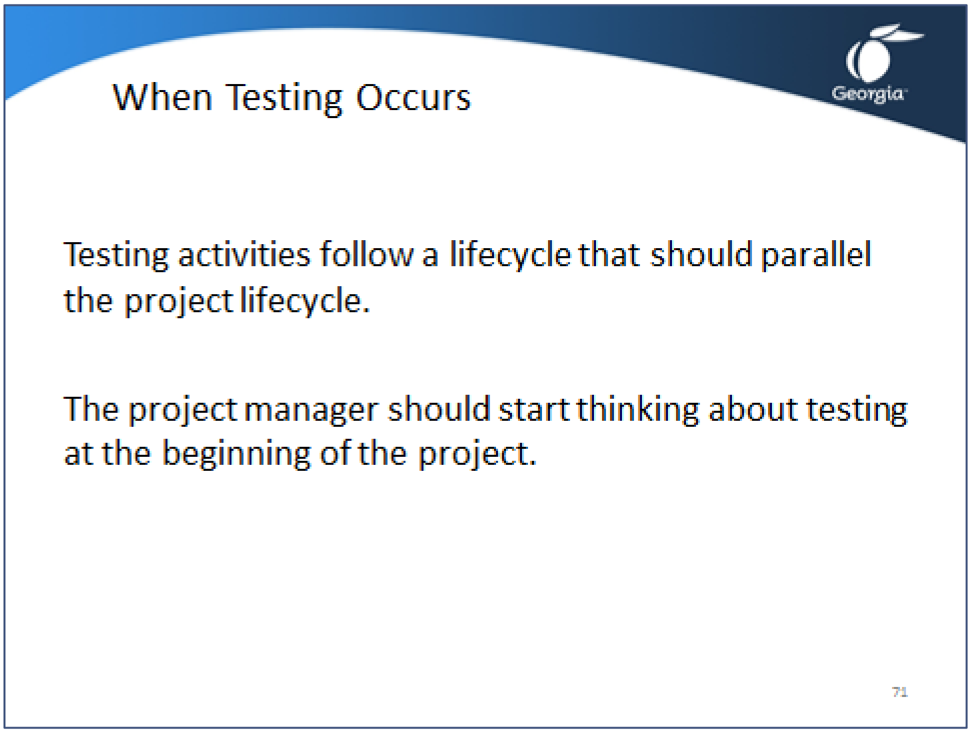
Test Script
The test script is a sequence of actions and expected results to accomplish a task. A test script is typically used to test a business process.

Validation Test Documentation

There are three main documents that are used by testers to plan and execute their work: the **test plan** and the **test case with test procedures (scripts)** as illustrated below.



Topic 2: When Testing Occurs



When Testing Occurs

Testing activities follow a lifecycle that should parallel the project lifecycle.

The project manager should start thinking about testing at the beginning of the project.

71

The Objective of Testing

The most important aspect of testing is to find defects. Testers find defects by looking for them, trying to find them, and expecting to find them.

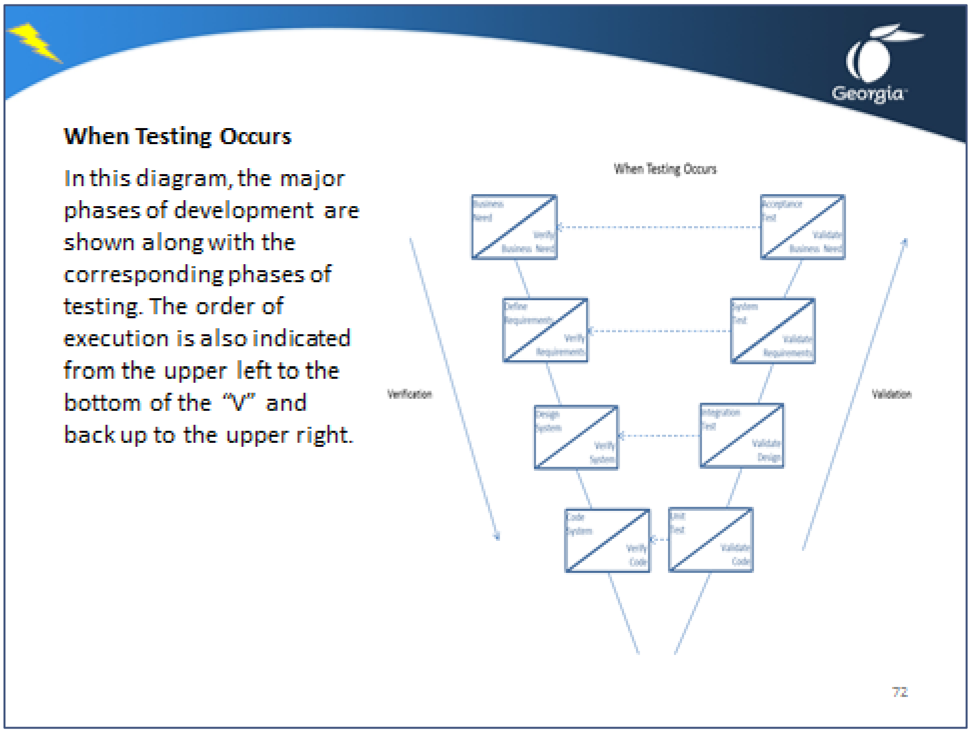
Defects are found by:

- Picking inputs to deliberately find bugs
- Trying to "break" the code
- Using independent testers

Testing activities follow a life cycle that should run parallel with the project life cycle. The project manager should start thinking about testing at the beginning of the project.

Software Development Life Cycle	Testing Phases
1. Plan the Project 2. Scope the Project	Test Planning
3. Elicit, Analyze and Document Requirements	Reviewing Requirements and Identifying Test Cases
4. Design a Solution	Writing Test Cases
5. Build or Buy the Solution	Test Preparation
6. Test the Solution: • Unit • Integration • System • User Acceptance	
7. Implement the Solution	
8. Conduct Post-Implementation Review	

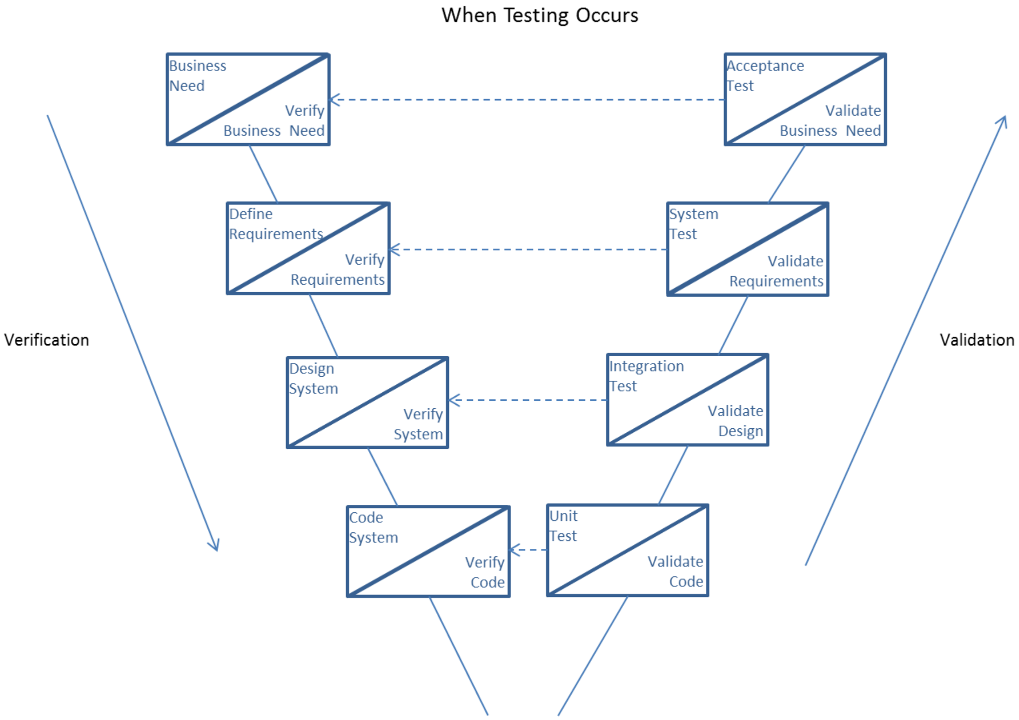
Topic 2: When Testing Occurs …continued



When Testing Occurs

In this diagram, the major phases of development are shown along with the corresponding phases of testing. The order of execution is also indicated from the upper left to the bottom of the "V" and back up to the upper right.

Georgia
When Testing Occurs
Business Need
Verify Business Need
Acceptance Test
Validate Business Need
Define Requirements
Verify Requirements
System Test
Validate Requirements
Verification
Validation
Design System
Verify System
Integration Test
Validate Design
Code System
Verify Code
Unit Test
Validate Code
72

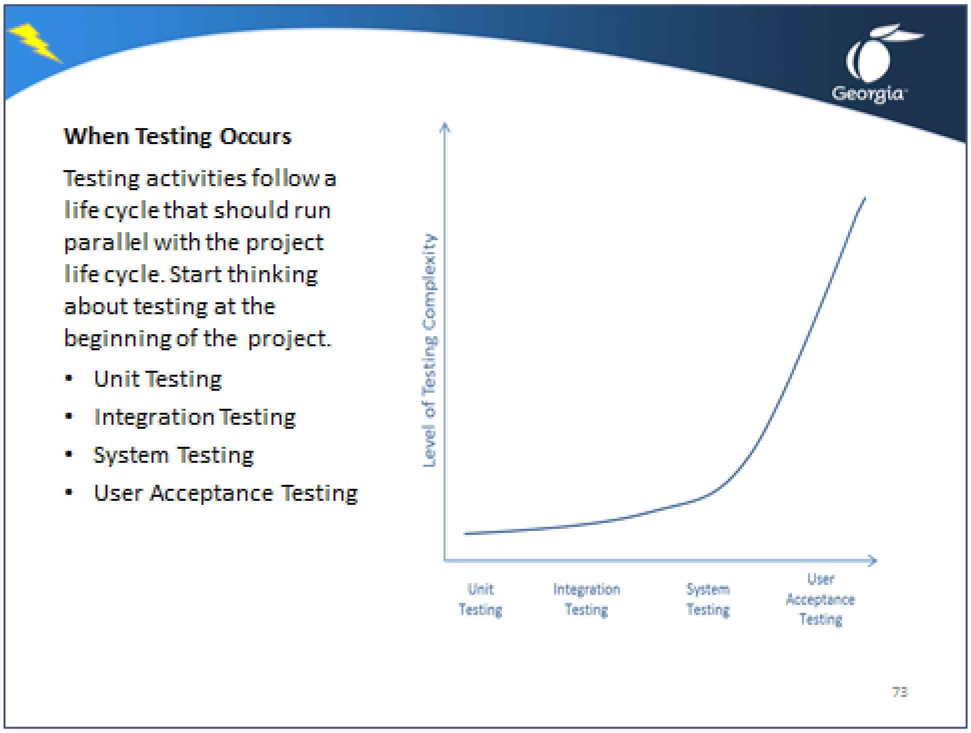


In this diagram, the major phases of development are shown along with the corresponding phases of testing. The order of execution is also indicated from the upper left to the bottom of the “V” and back up to the upper right.

Each box has either a verification step (a test performed by inspecting something) or a validation step (a test performed by executing software on a computer).

The “V” diagram can be used to depict a testing methodology. It is important to note that user acceptance testing is at the top of the “V” and validates business or operational need, not that the system was built according to requirements.

Topic 2: When Testing Occurs …continued



When Testing Occurs

Testing activities follow a life cycle that should run parallel with the project life cycle. Start thinking about testing at the beginning of the project.

- Unit Testing
- Integration Testing
- System Testing
- User Acceptance Testing

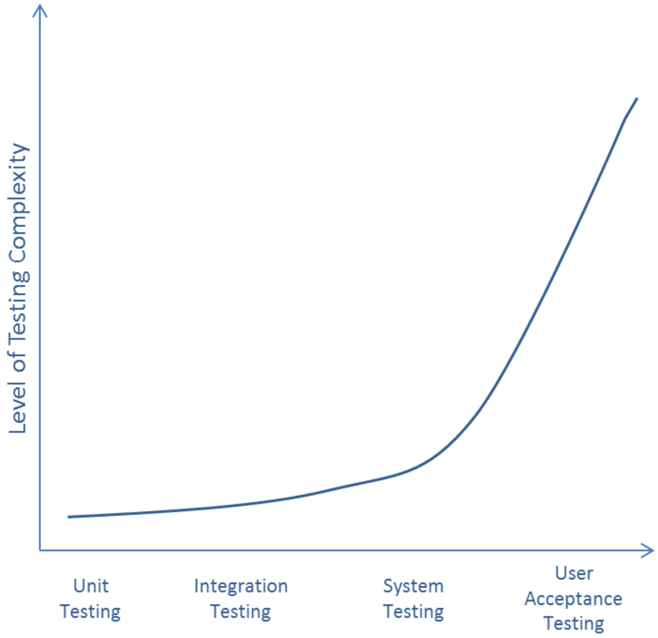
Types of Validation Testing

- **Unit Testing** is the first phase of testing. It involves testing each "unit" built for the system as a stand-alone test. The development team performs unit testing and attempts to find problems in the smallest components of the system before testing it in its entirety.
- **Integration Testing** is the second phase of testing and requires the individually tested "units" to be integrated together and tested as a larger unit. The development team or testing team performs integration testing attempting to find problems in how components of the system work together.
- **System Testing** is the third phase of testing. The testing team attempts to find problems in how the software meets the users' needs and validates that the software meets the original requirements.
    - **Regression testing** is also a part of system testing. In regression testing the team will "re-test" the software after changes are made to validate the changes did not "break" components already tested.
    - **Dynamic testing includes:**
        - Performance testing – how fast the system can complete a function
        - Stress testing – pushing the system to the limits in terms of number of users, rate of input and speed of response
        - Volume testing – testing high volume transactions to verify that the software will handle all growth projections
    - **Security testing** – making sure that unauthorized users cannot gain access and that authorized users can effectively complete their required tasks
    - **Installation testing** – important for software that will be shipped to users and will require an installation procedure

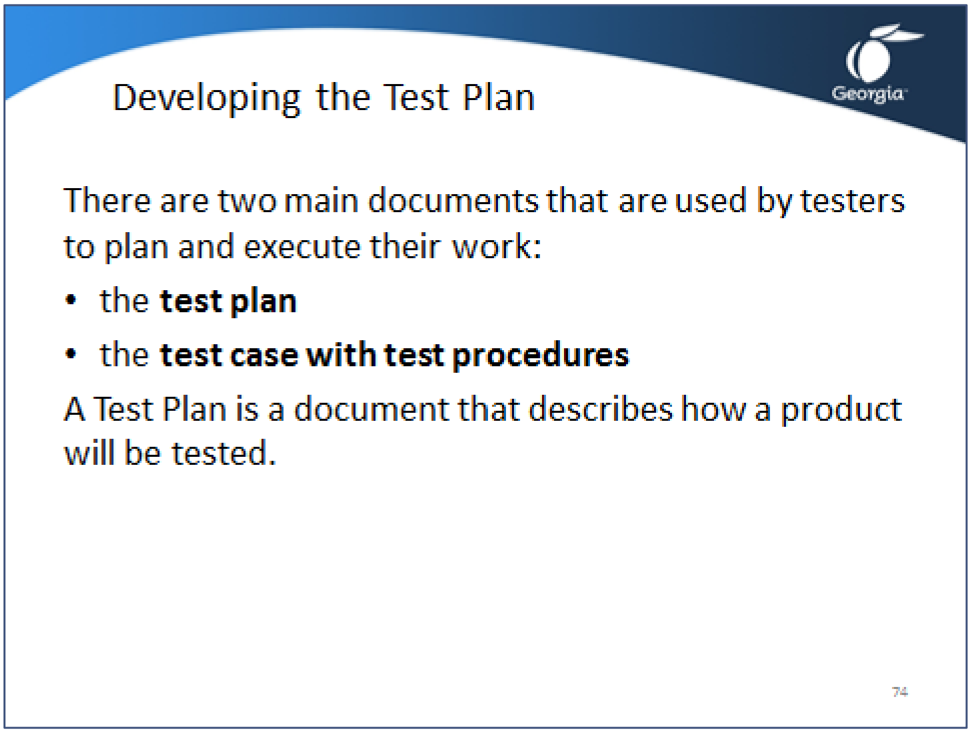
    - **Configuration testing** – determining how will the software perform on various types of hardware, operating systems, networks; and in conjunction with other software packages running on the same system
- **User Acceptance Testing** is the final phase of testing. Users test real-life scenarios on the software to verify that it will meet their needs. The users perform the tests attempting to show compliance, not find bugs.

  User Acceptance Process:
    - Get an agreement with the customer/user on test cases and pass criteria
    - Be open about known defects
    - Have the customer/user participate in the testing
    - Ask for signoff (required before Implementation can occur)

The Complexities of Testing



Topic 3: Developing the Test Plan



Developing the Test Plan

There are two main documents that are used by testers to plan and execute their work:

- the **test plan**
- the **test case with test procedures**

A Test Plan is a document that describes how a product will be tested.

74

A Test Plan is a document that describes how a product will be tested. There are several sections in a test plan:

- Introduction
- Test items
- Features to be tested
- Features not to be tested
- Approach
- Item pass/fail criteria
- Suspension and resumption criteria
- Test deliverables
- Testing tasks
- Environmental needs
- Responsibilities
- Staffing and training needs
- Schedule
- Risks and contingencies
- Approvals

Sample Test Plan

Test Plan ID	Test Plan Name	Date

Introduction:

Summarize the software items and software features to be tested. The need for each item and its history may be included.

Reference to the following documents, when they exist, are required in the highest level test plan:

a) Project authorization
b) Project plan
c) Quality assurance plan
d) Configuration management plan
e) Relevant policies
f) Relevant standards

Test Items:	Identify the test items including their version/revision level. Also specify characteristics of their transmittal media that impact hardware requirements or indicate the need for logical or physical transformations before testing can begin (e.g., programs must be transferred from tape to disk). Supply the references to the following test item documentation, if it exists; a) Requirements specification b) Design specification c) Users guide d) Operations guide e) Installation guide Reference any incident reports relating to the test items. Items that are to be specifically excluded from testing may be identified.
Features to be Tested:	Identify all software features and combinations of software features to be tested. Identify the test design specification associated with each feature and each combination of features.
Features *not* to be Tested:	Identify all features and significant combinations of features that will *not* be tested and the reasons.
Approach:	Describe the overall approach to testing. For each major group of features or feature combinations, specify the approach that will ensure that these feature groups are adequately tested. Specify the major activities, techniques, and tools that are used to test the designated groups of features. The approach should be described in sufficient detail to permit identification of the major testing tasks and estimation on the time required to do each one. Specify the minimum degree of comprehensiveness desire. Identify the techniques that will be used to judge the comprehensiveness of the testing effort (e.g., determining which statements have been executed at least once). Specify any additional completion criteria (e.g., error frequency). The techniques to be used to trace requirements should be specified. Identify significant constraints on testing such as test item availability, testing resource availability, and deadlines.
Item Pass/Fail Criteria:	Specify the criteria to be used to determine whether each test item has passed or failed testing.
Suspension criteria and resumption:	Specify the criteria used to suspend all or a portion of the testing activity on the test items associated with this plan. Specify the testing activities that must be repeated when testing is resumed.
Test Deliverables:	Identify the deliverable documents. The following documents should be included: a) Test plan b) Test design specifications c) Test case specifications d) Test procedure specifications

<table>
<tr><td rowspan="1"></td><td colspan="3">e) Test item transmittal reports
f) Test logs
g) Test incident reports
h) Test summary reports

Test input data and test output data should be identified as deliverables.

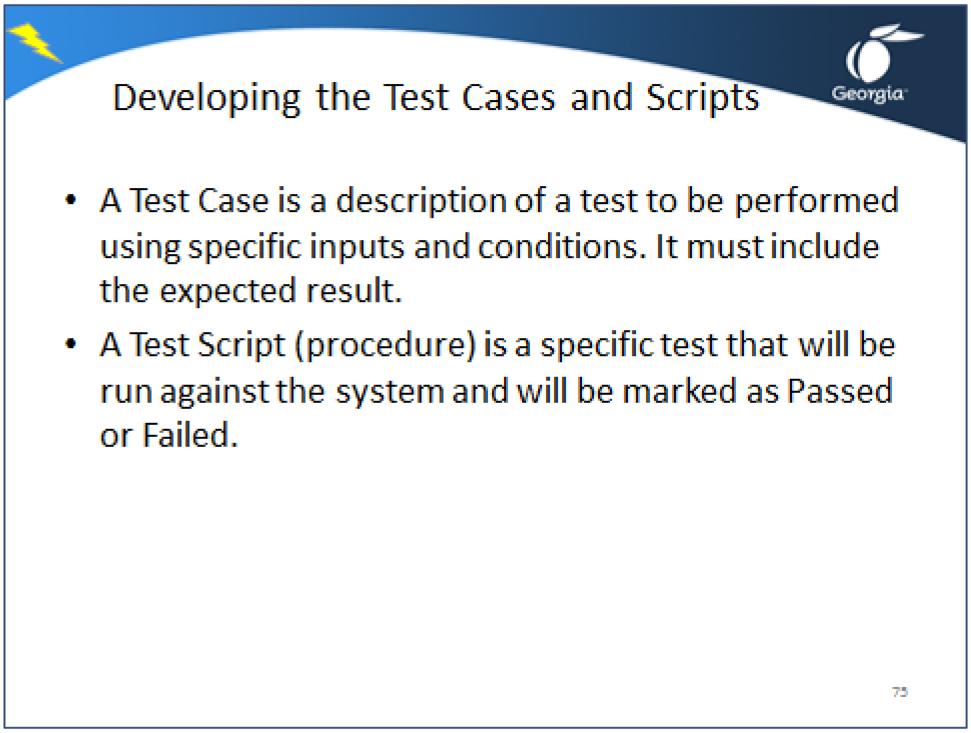
Test tools may also be included.</td></tr>
<tr><td><b>Testing tasks:</b></td><td colspan="3">Identify the set of tasks necessary to prepare for and perform testing. Identify all intertask dependencies and any special skills required.</td></tr>
<tr><td><b>Environmental needs:</b></td><td colspan="3">Specify both the necessary and desired properties of the test environment. This specification should contain the physical characteristics of the facilities including the hardware, the communications and system software, the mode of usage (e.g., stand-alone), and any other software or supplies needed to support the test. Also specify the level of security that must be provided for the test facilities, system software, and proprietary components such as software, data, and hardware.

Identify special test tools needed. Identify any other testing needs (e.g., publications or office space). Identify the source for all needs that are not currently available to the test group.</td></tr>
<tr><td><b>Responsibilities:</b></td><td colspan="3">Identify the groups responsible for managing, designing, preparing, executing, witnessing, checking, and resolving. In addition, identify the groups responsible for providing the <b>Test Items</b> and the <b>Environmental Needs</b>.

These groups may include the developers, testers, operations staff, user representatives, technical support staff, data administration staff, and quality support staff.</td></tr>
<tr><td><b>Staffing and training needs:</b></td><td colspan="3">Specify test staffing needs by skill level. Identify training options for providing necessary skills.</td></tr>
<tr><td><b>Schedule:</b></td><td colspan="3">Include test milestones identified in the software project schedule as well as all item transmittal events.

Define any additional test milestones needed. Estimate the time required to do each testing task. Specify the schedule for each testing task and test milestone. For each testing resource (i.e., facilities, tools, and staff), specify its periods of use.</td></tr>
<tr><td><b>Risks and Contingencies:</b></td><td colspan="3">Identify the high-risk assumptions of the test plan. Specify contingency plans for each (e.g., delayed delivery of test items might require increased night shift scheduling to meet the delivery date).</td></tr>
<tr><td rowspan="3"><b>Approvals:</b></td><td><b>Req’d</b></td><td colspan="2"><b>Signature and Date</b></td></tr>
<tr><td></td><td></td><td>Date:</td></tr>
<tr><td></td><td></td><td>Date:</td></tr>
</table>

Topic 4: Developing Test Cases and Scripts



Developing the Test Cases and Scripts

- A Test Case is a description of a test to be performed using specific inputs and conditions. It must include the expected result.
- A Test Script (procedure) is a specific test that will be run against the system and will be marked as Passed or Failed.

75

A Test Case is a description of a test to be performed using specific inputs and conditions. It must include the expected result. The project team will develop hundreds, if not, thousands of test cases to support the testing effort.

A Test Script (procedure) is a specific test that will be run against the system and will be marked as Passed or Failed. A single test case may be executed by one or many test scripts (procedures).

Each test case/script document describes, in detail:

- Test items
- Input specifications
- Output specifications
- Environmental needs
- Special procedural requirements
- Procedure steps
- Expected Results
- Inter-case dependencies
- Approvals

Sample Test Case/Script

Test Case ID (unique identifier)	Test Case Name	Date:

Test Items:

Identify and briefly describe the items and features to be exercised by this test case. For each item consider supplying references to the following test item documentation:

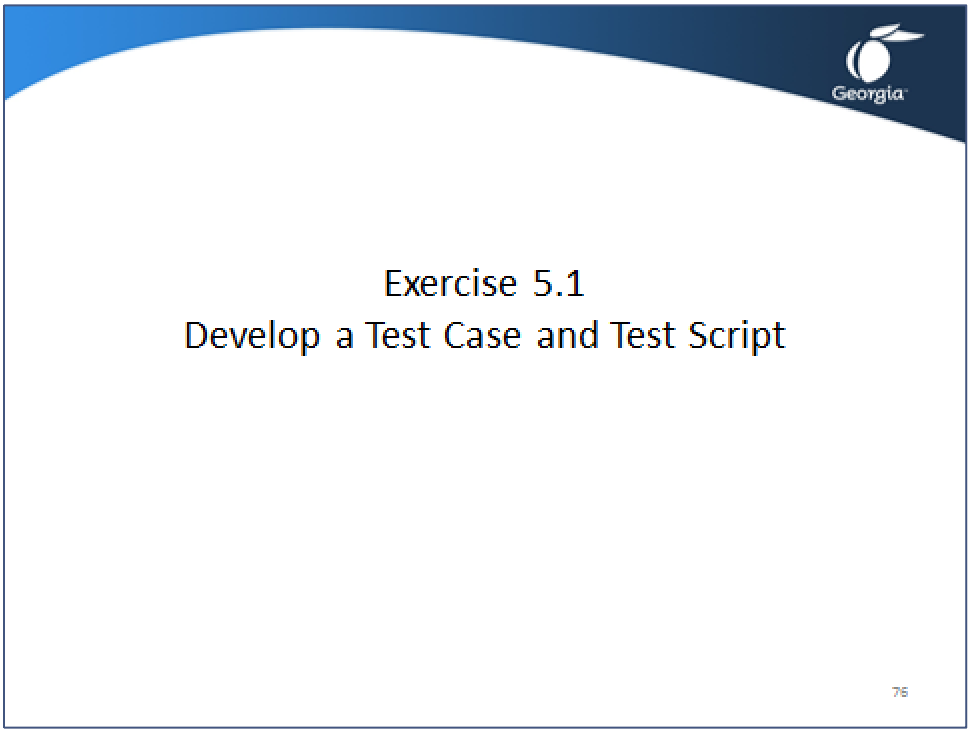
a) Requirements specification
b) Design specification
c) Users guide
d) Operations guide
e) Installation guide

Input specifications:	Specify each input required to execute the test case. Some of the inputs will be specified by value (with tolerances where appropriate), while others, such as constant tables or transaction files, will be specified by name. Identify all appropriate databases, files, terminal messages, memory resident areas, and values passed by the operating system. Specify all required relationships between inputs (e.g., timing).
Output specifications:	Specify all of the outputs and features (e.g., response time) required of the test items. Provide the exact value (with tolerances where appropriate) for each required output or feature.
Environmental needs:	Hardware: Specify the characteristics and configurations of the hardware required to execute this test case (e.g., 132 character, 24-line CRT). Software: Specify the system and application software required to execute this test case. This may include system software such as operating systems, compilers, simulators, and test tools. In addition, the test item may interact with the application software. Other: Specify any other requirements such as unique facility needs or specially trained personnel.
Special procedural requirements:	Describe any special constraints on the test procedures that execute this test case. These constraints may involve special set up, operator intervention, output determination procedures, and special wrap up.
Inter-case dependencies:	List the identifiers of the test cases that must be executed prior to this test case. Summarize the nature of the dependencies.

Procedure Steps	Expected Results	Pass/Fail

Approvals:	Req'd	Signature and Date	
			Date:
			Date:

Exercise 5.1 Develop a Test Case and Test Script



Exercise 5.1

Develop a Test Case and Test Script

Develop test cases, scripts for the case study project using the template below. Also, update the Requirements Traceability Matrix so that the requirements correspond to the developed test case.

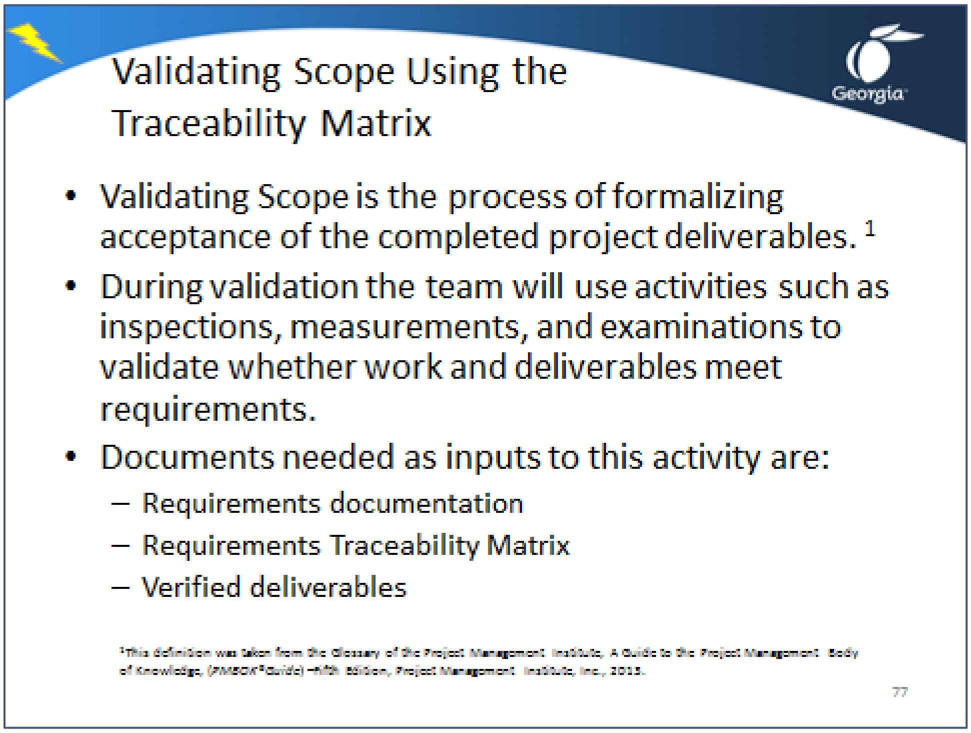
Speedy Office Supplies Web Expansion Project Test Case/Script

Test Case ID		Date:
Test Items:		
Input specifications:		
Output specifications:		
Environmental needs:		
Special procedural requirements:		
Inter-case dependencies:		

Procedure Steps	**Expected Results**	**Pass/Fail**

Approvals:	**Req'd**	**Signature and Date**	
			Date:
			Date:

Topic 5: Validating Scope Using the Traceability Matrix



Validating Scope Using the Traceability Matrix

- Validating Scope is the process of formalizing acceptance of the completed project deliverables. [1]
- During validation the team will use activities such as inspections, measurements, and examinations to validate whether work and deliverables meet requirements.
- Documents needed as inputs to this activity are:
  - Requirements documentation
  - Requirements Traceability Matrix
  - Verified deliverables

[1] This definition was taken from the Glossary of the Project Management Institute, A Guide to the Project Management Body of Knowledge, (*PMBOK*®Guide) –Fifth Edition, Project Management Institute, Inc., 2013.

77

Validating Scope is the process of formalizing acceptance of the completed project deliverables. [1] The key benefit of this activity is that it brings objectivity to the acceptance processes and increases the chance of final product, service, or result acceptance by validating each deliverable (requirement). [2]

Documents needed as inputs to this activity are:

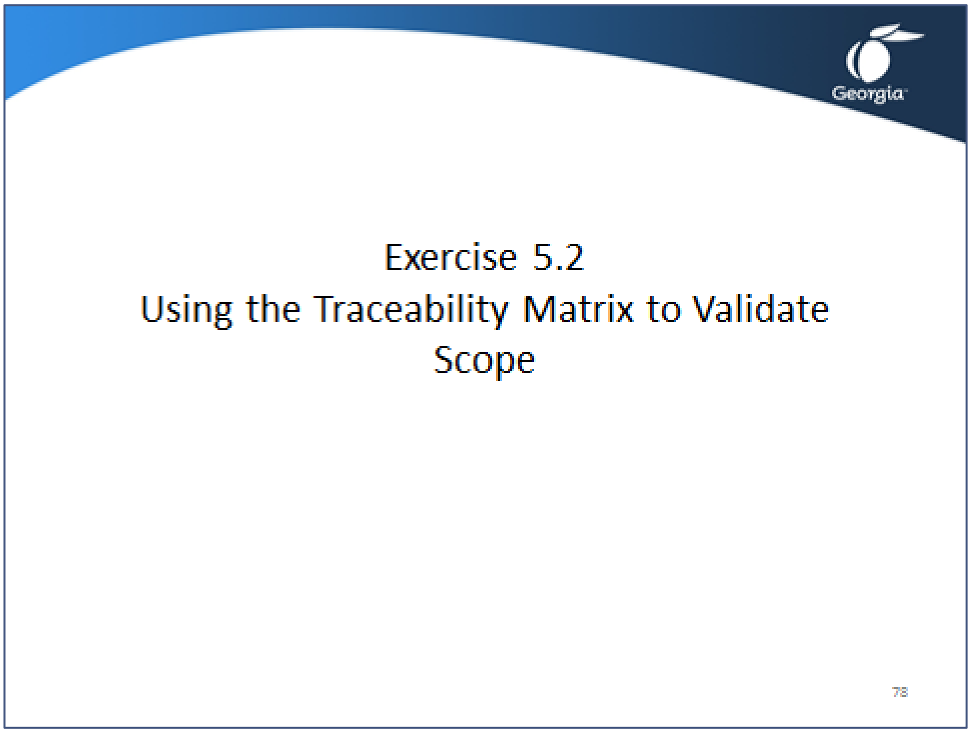
- Requirements documentation
- Requirements Traceability Matrix
- Verified deliverables

Deliverables that have been checked for correctness through quality control processes are now reviewed with the customer to ensure that they are completed satisfactorily and have received formal acceptance by the customer. The requirements documentation and the requirements traceability matrix are used to perform the validation of the scope.

These activities differ from quality control in that they are concerned with acceptance of the deliverables, while quality control is concerned with the correctness of the deliverables. During validation the team will use activities such as inspections, measurements, and examinations to validate whether work and deliverables meet requirements. [3]

[1] This definition was taken from the Glossary of the Project Management Institute, A Guide to the Project Management Body of Knowledge, (*PMBOK®Guide*) –Fifth Edition, Project Management Institute, Inc., 2013.
[2] *PMBOK® Guide*, Page 133
[3] *PMBOK® Guide*, Page 134

Exercise 5.2 Using the Traceability Matrix to Validate Scope



Exercise 5.2
Using the Traceability Matrix to Validate Scope

78

Using the Requirements document, Use Cases, Test cases/scripts, and the Requirements Traceability Matrix from prior exercises, discuss below how you would use the Requirements Traceability Matrix to validate the requirements and scope of the Speedy Office Supplies Web Expansion Project.

Lesson 5 Summary: Learning Objectives Recap

- **Discuss testing terminology**
  Verification is “the evaluation of whether or not a product complies with a requirement.
  Validation is “the assurance that a product, service, or system meets the needs of the customer
  Test Plan is a document specifying the scope, approach, resources, and schedule of test activities.
  Test Case is a standalone document that specifies inputs (conditions), predicted results, and a set of execution steps (procedures) for each item.
  Test Script is a sequence of actions and expected results to accomplish a task.

- **Discuss various testing methods to validate requirements**
  Unit Testing involves testing each “unit” built for the system as a stand-alone test.
  Integration Testing requires the individually tested “units” to be integrated together and tested as a larger unit.
  System Testing attempts to find problems in how the software meets the users’ needs
  User Acceptance Testing allows users to test real-life scenarios on the software to verify that it will meet their needs.

- **Recognize how to develop a test plan for gathered requirements**
  A Test Plan is a document that describes how a product will be tested. There are several sections in a test plan:
  - Introduction
  - Test items
  - Features to be tested
  - Features not to be tested
  - Approach
  - Item pass/fail criteria
  - Suspension and resumption criteria
  - Test deliverables
  - Testing tasks
  - Environmental needs
  - Responsibilities
  - Staffing and training needs
  - Schedule
  - Risks and contingencies
  - Approvals

- **Explain how to develop Test Cases and Scripts to validate requirements**
  A Test Case is a description of a test to be performed using specific inputs and conditions. A Test Script (procedure) is a specific test that will be run against the system and will be marked as Passed or Failed.

  Each test case/script document describes, in detail:
  - Test items
  - Input specifications
  - Output specifications
  - Environmental needs
  - Special procedural requirements
  - Procedure steps
  - Expected Results
  - Inter-case dependencies
  - Approvals

- **Explain how to validate requirements and scope using the traceability matrix**
  The requirements documentation and the requirements traceability matrix are used to perform the validation of the scope.

Notes

CASE STUDY – SPEEDY OFFICE SUPPLIES WEB EXPANSION PROJECT

Company Overview

Speedy Office Supplies has been in business for 15 years and is recognized as the leader in discount office supplies. We have a reputation of providing high quality products at reasonable prices and offering superior customer service. We are selling to corporate clients, governmental agencies, and individuals. Our customers are served by over 40,000 employees through direct sales, catalogs, e-commerce and more than 2,000 stores. Eighty percent of our business is currently done in our 2,000 retail stores.

Over the past five years the Retail Store Division has shown a steady decline in sales and profitability; energy costs have increased by 30% for our fleet vehicles and retail stores; employee health care costs have increased by 75% and continue to rise. Market trends and customer preferences are indicating that customers desire the ability to order their products on-line at times convenient to them. The SOS management team believes if we phase-out the Retail Store Division and replace it with a web-based ordering system and consolidation of our distribution network, we anticipate a savings of nearly 10 million dollars per year. This would also need to integrate into the existing supply chain systems. Customer satisfaction surveys also indicate a favorable reaction to the concept of web-based sales, which could increase our current sales by at least 25% over the next 5 years. Based on this information SOS management has made a decision to close all the brick and mortar stores within 18 months. We believe this decision will significantly cut costs and that we can be just as successful selling our products on our website.

Currently orders for products are received via in-store requests, phone calls, or catalog mail-in from customers. We access our online system to check inventory, prices, and estimated shipping dates. If the order total is over $100,000 we turn it over to a supervisor. We then call the Credit Card Authorization Company to check the customer’s credit card account. If the credit card charge is authorized we enter the order into the system. The current system is an old mainframe application and is very cumbersome.

There are purchasing agreements, special discounts, and payment terms for our clients purchasing over $50,000 per year. In the past, we have billed these customers on a monthly basis, providing them with a detailed listing by location of their purchases. We want to make it easier for them to pay via credit card each time they place an order to increase our cash flow and lower our Accounts Receivable. If possible, we still want to provide select customers the same reporting on a monthly basis for their purchases by location.

Federal Express and UPS are currently bidding on the exclusive rights for delivery of all customer office supplies. Each company is proposing an online interface to track shipments, including the name of the person who signs for the delivery. The shipment will need to have a label and detailed purchase order slip with the package. The cost of shipping is determined by the size of the package, weight, location, insurance, and timeliness of delivery. The customer will need an accurate shipping cost at the time of purchase.

Project Request

Our main focus for this project is to create the shopping experience for our retail customer on the website and to place product orders on the Internet. We want to have real time information regarding product description; quantities; pricing; availability; payment processing; shipping method options with associated costs; delivery date; and order tracking. All information currently available at the retail stores and in the catalogs should be available and consistent with the Internet.

It would be nice if there were a place on the Internet for the customer to build a profile and store frequently purchased items in a list to use for future purchases. This would be very beneficial for large organizations that purchase the same products frequently.

We envision using our existing customer number and allowing each customer to create a password to ensure security. Anyone could look at the products online, but only registered customers would be allowed to place orders. The web site should have search ability by several options: product item number (from the catalog), product type, color, and size.

Hopefully when a customer places an order the software would quickly calculate a shipping charge and present the order total to the customer. We would not allow orders totaling more than $1000 to be placed on the web. The software should also email a confirmation to the customer if requested.

Departments Involved

The **Marketing Department** is responsible for customer reporting and the negotiations for preferred customer status including volume discounts. Our largest customers receive one monthly bill for all their departments’ purchases and a report showing the detailed purchases. Additionally, marketing maintains the customer profiles, which are used to process orders, verify billing information, discounts, and reduce redundancy by eliminating the need for the customer to always enter their company information.

The **Customer Service Department** will need access to all information regarding customer orders to assist with the web site usage and handle any possible complaints.

Accounts Receivable is responsible for processing and sending bills to our preferred customers. The web ordering system will need to notify accounts receivable when one of our preferred customers request their order to be direct billed. Some customers have negotiated payment terms and discount rates based on volumes. They work with the Collections Department for any outstanding receivables beyond 90 days. On a monthly basis Accounts Receivable produces an aging report.

Inventory Management is impacted by a reduction in inventory from placed orders and an increase in inventory from cancellations and returns. They are responsible for managing the inventory and placing orders with vendors. Inventory Management is also responsible for handling returns, including items that have to be returned to the suppliers as defective.

Order Fulfillment receives an order notification from the order processing system containing all necessary information required to assemble the order. They are responsible for producing the packaging slips, retrieving the supplies, assembling the order into a bin or crate, and delivering the order to the Shipping Department.

The **Shipping Department** receives the order from fulfillment and prepares the order for shipment. The packing slip contains the shipping method requested by the customer and the estimated shipping timeframe. The Shipping Department is responsible for notifying the shipping company and updating the order status.

Outside Organizations

The **Shipping Company** currently has an online tracking system. Our web ordering system will have a direct link to the shipping company's web site for the customer to track packages using the tracking number provided by the Shipping Department to the order status system.

The **Credit Card Processor** currently authorizes customer purchases made in the stores, over the phone, or via fax. An additional interface will need to be established between the web application to receive the customer and order information and to return an authorization code.

APPENDIX I - EXERCISE ANSWERS

Exercise 1.1 Evaluating Requirements

Review each requirement below and describe why each requirement is not excellent.

Requirement
"A customer is any company that has done business with our organization in the past or potentially will in the future."

Why is this requirement not Excellent? **Too ambiguous.**

A customer is any organization that has purchased one or more of our products since January 1, 1985.

Requirement
""For each of our customers, we need to know the names of each employee with which our organization has contact."

Why is this requirement not Excellent? **Too vague.**

The system must record a unique contact record containing First Name, Last Name, Phone Number, email address, mailing address, city, state, zip code, and contact notes for each contact made by an employee of the company.

Requirement
"Telemarketers should receive the next phone call on their screen as quickly as possible after ending the previous call."

Why is this requirement not Excellent? **Not Verifiable.**

Telemarketers must receive the next phone call on their screen within 1.5 seconds after the previous call is disconnected.

Exercise 1.2 Develop the Project Charter

Develop the Project Charter for the Case Study project using the components listed below.

Project Charter Components

Project Purpose or Justification

The migration of order entry functions from the legacy mainframe system to the web-based platform will result in greater efficiency with regards to company resources and business processes. Moving to a web-based ordering platform will enable Speedy Office Supply to manage its order entry, and supply chain functions in a seamless and consolidated manner.

Project Objectives

Timely and accurate order fulfillment, Improve operating efficiency, Reduce overhead costs, Increase sales, and Increase customer satisfaction.

Project description

The Web Expansion Project will review and analyze several potential products to replace Speedy Office Supply's legacy system with a web-based platform. This will be done by determining and selecting a product which adequately replaces our existing system.

High-level Requirements

Web-based order entry and processing system that allows customer to scan product listings, order selected products, confirm order and set up a profile. System must interface with current legacy inventory and accounts receivable systems and external credit card and shipping systems for authorization and order tracking.

High-level risks

The following items are seen as initial high-level risks:

- Invalid or incomplete requirements,
- Resources not available,
- New system is incompatible with legacy system,
- Functionality between new and legacy system is missing.

Project manager, authority level

John Q. Doe is the project manager. His level of authority includes the development and execution of the project plan with managerial oversight of designated resources allocated to the project.

Exercise 2.1 Identify Customer Needs

Using the information provided in the Case Study, determine the customer needs using the Needs Lifecycle.

Needs Emergence:

Speedy Office Supplies is faced with competing against other office supply retailers using the internet to reduce cost of operations and expand their customer base. Speedy Office Supply needs to reduce its operation cost in the brick and mortar stores and increase their sales.

Needs Recognition:

Speedy recognizes that in order to reach their objectives of reducing operational expenses and increasing sales through expanding the customer base the company will move to close the brick and mortar retail store by the 4th quarter of the next year and migrate to an on-line retail model.

Needs Articulation:

Speedy Office Supply will initiate a project to create a web-based retail shopping experience while implementing a phased approach to closing the brick and mortar retail stores.

Exercise 2.2 Identify Stakeholders and Perform Stakeholder Analysis

Stakeholder Register

SH ID	Stakeholder Name	Project Role	Major Requirements	Main Expectations
SH001	Retail Customer	Customer	Browse and select product from Web site, select payment method and pay online	Improve the retail shopping experience. Fast order entry and prompt shipping.
SH002	Sam Speedy	CEO, Sponsor	Create retail shopping experience and close retail stores	Project completes on time and on budget. Customer satisfaction increases.
SH003	Inventory Management	SME	Need new system to provide legacy Inventory system with appropriate information	Data from new system is accurate and updates are fast
SH004	Credit Card Processor	Provider	Need credit card info from customer to credit accounts	Expect accurate and fast transactions
SH005	Shipping Company	Provider	Tracking numbers provided to shipping dept match new system	Accurate tracking numbers and fast access to system
SH006	Retail Store employees	Employee	Kept informed about the store closing and impact	Treated fairly by Speedy Office Supplies

Stakeholder Analysis

SH ID	Stakeholder Name	Impact	Influence	Interest	Power	Urgency	Legitimacy
SH001	Retail Customer	3	3	3	2	3	3
SH002	Sam Speedy	3	3	2	3	2	3
SH003	Inventory Management	3	2	3	2	2	2
SH004	Credit Card Processor	1	1	2	1	1	2
SH005	Shipping Company	1	1	2	1	1	2
SH006	Retail Store employees	2	2	2	1	1	2

3= High
2= Medium
1= Low

Impact: The SH ability to effect changes to the project's execution or planning.
Influence: The SH active involvement in the project.
Interest: The SH level of interest regarding project outcomes.
Power: The SH ability to impose their will on the project.
Urgency: The SH need for immediate attention.
Legitimacy: The SH involvement is appropriate.

Exercise 3.1 Identifying Core Components of Requirements

Using the information provided in the Case Study identify the core components of the requirements.

ENTITIES/ATTRIBUTES (DATA):
STORE: StoreID, StoreName, StoreAddress, StoreManager, StorePhone
CUSTOMER: CustID, CustName, CustAddress, CustPhone, CustCrdCard,
ORDER: OrderID, OrderCustID, OrderDate, OrderShpTo,
ORDERLINE: OrderLineID, OrderLine#, OrderLineProdID, OrderLineQty, OrderLinePrice

PROCESSES:
Place an Order
Login Customer
Create Customer Profile

BUSINESS RULES:
A customer may order any item from the catalog
A customer must login with a user id and password to place an order
An order exceeding $1000 cannot be placed on the web-based system
An order exceeding $100,000 must be approved by a manager

EXTERNAL AGENTS or ACTORS:
Customer
Shipping Clerk
Customer Service Representative
Credit Card Company
Shipping Company

Exercise 3.2 Identifying Product Scope

Using the information provided in the Case Study and the table on the following page identify the product scope.

PRODUCT SCOPE PLANNING WORKSHEET

Unique Number	Description (i.e. feature or function)	Business Priority	Technical Priority	Estimate (cost, time)	Phase, release, iteration
1	Allow customers to place orders online	M	H	H	P1
2	Allow customers to request items not in the catalog	C	L	M	R1
3	Add a wish list for customers to maintain future desired orders	S	M	M	P1
4	Build a chat function to answer customer questions online	S	M	M	P1
5	Add an option to request a catalog	C	L	L	R1

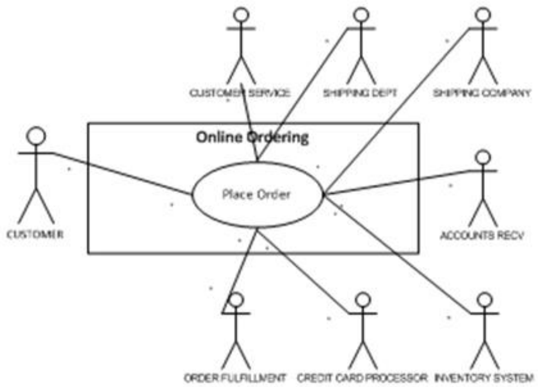
Exercise 3.3 Requirements Traceability Matrix

Using the information provided in the Case Study and the prior exercise begin to fill in the Requirements Traceability Matrix.

Requirements Traceability Matrix							
Project Name:	Speedy Office Supplies Web Expansion Project						
Project Description:	Create the retail shopping experience for customers on our website.						
Cost Center:	Customer Service Department						
ID	Requirements Description	Use Case ID	Use Case Name	Test Case ID	Test Case Name	Date Tested	Approved By
R1	The system must allow customer access to existing catalog of products and browse current offerings.						
R2	The system must allow the customer to place an order by selecting the product id and quantity desired.						
R3	The system must interface to the legacy inventory system to retrieve product description, pricing, and availability information.						
R4	The system must allow the customer to add or remove items from the order.						
R5	The system must interface with the credit card provider to authorize the customer order and provide an authorization code.						
R6	The system must interface with the legacy order fulfillment system and provide order fulfillment information.						
R7	The system must provide shipping options to the customer from the legacy shipping system.						

Exercise 3.4 Developing the Use Case

Using the information provided in the Case Study, the Use Case Description template below, and Exercises 3.1 and 3.2 develop a use case for the Place an Order process. Also, update the Requirements Traceability Matrix.



Use Case Template

Use Case ID	UC001	
Use Case Name	Place an Order	
Primary Path		
Actor Actions	**System Responses**	
1. Customer selects one or more products	2. System retrieves product information for display	
3. Customer provides profile, shipping, and payment information	4. System confirms information and displays order information to confirm with customer	
5. Customer confirms order	6. System sends order to order fulfillment, displays shipping and confirmation information	
7. Customer exits or returns to browsing		
Alternate Path 1	Describes a sequential flow of actions related to a specific condition, under which the system exhibits a different behavior.	
Actor Actions	**System Responses**	
1. Invalid Credit Card is used	2. System displays error message	
2. Product not in stock	3. System displays not in stock message	
Post Conditions	No post conditions	
Additional Notes		
Revision History	**Revised by**	**Date**

Requirements Traceability Matrix

Requirements Traceability Matrix							
Project Name:	Speedy Office Supplies Web Expansion Project						
Project Description:	Create the retail shopping experience for customers on our website.						
Cost Center:	Customer Service Department						
ID	Requirements Description	Use Case ID	Use Case Name	Test Case ID	Test Case Name	Date Tested	Approved By
R1	The system must allow customer access to existing catalog of products and browse current offerings.	UC001					
R2	The system must allow the customer to place an order by selecting the product id and quantity desired.	UC002					
R3	The system must interface to the legacy inventory system to retrieve product description, pricing, and availability information.	UC003					
R4	The system must allow the customer to add or remove items from the order.	UC004					
R5	The system must interface with the credit card provider to authorize the customer order and provide an authorization code.	UC005					
R6	The system must interface with the legacy order fulfillment system and provide order fulfillment information.	UC006					
R7	The system must provide shipping options to the customer from the legacy shipping system.	UC007					

Exercise 4.1 Develop Non-Functional Requirements

Using the information provided in the Case Study and the template provided on the next page, develop the non-functional requirements.

Reqmt number	Requirement	Category/type
PF001	The system shall be able to process 100 order transactions per second in peak load.	Performance
SC001	The length of a customer password must be between 8 and 15 characters and consist of at least one upper case character, one symbol, and one number.	Security
RL001	The online order entry system shall be available from 6:00 am EST to 11:00 pm EST.	Reliability
CM001	The system must be able to interface with any HTML browser.	Compatibility
MN001	The development process must have a regression test procedure which allows complete re-testing within two business days.	Maintainability
TR001	The system shall be developed for MS Windows® and Microsoft® operating system platforms.	Transferability
US001	An experienced user must be able to update a customer order within 2 minutes.	Usability

Exercise 4.2 Develop Transitional Requirements

Determine the transitional requirements for the project case study.

- Plan the cutover/implementation of software and hardware
- Develop a rollout plan
- Turnover the solution to the operational team

Cutover Plan:

The web site will be introduced after a pilot to a selected region. After the pilot the site will be rolled out nationally. The retail stores will remain open while the new system is piloted. After the pilot, stores will be closed in a phased approach according to a close-out plan and schedule.

Rollout Plan:

Date/Time	Task	Responsible Person	Back-out/contingency
April 3 – Friday at noon	All users log out of the system after finishing the weeks data entry work.	John – Business area representative	If work cannot be completed, keep paper backup of information so that it can be entered into the new system after the cutover.
April 3 – Friday 12:30 – 1:30 pm	Backup database. Print pre-conversion counts.	Jeff – IT DBA	Backup has been tested and timed at about 45 minutes.
April 3 – Friday 1:30 – 5 pm	Convert production data to new database format.	Donna – IT Dev	Conversion programs have been tested and timed at about 2 hours.
	Review error report from conversion and make corrections.	Donna – IT Dev John – Business	Data cleanup should take no longer than 1 hour.
	Finalize conversion. Print post conversion counts.	Donna – IT Dev	Pre-conversion counts should match post-conversion counts. If not – find discrepancy and correct if possible. If more than 10% errors, rollout will be cancelled and rescheduled for future.

Turnover:

The change management plan must include turnover to the operational support group. This turnover may include:

+ Develop Service Level Agreements (SLA). These are based on non-functional requirements and their measurements
  Example:

  Non-functional requirement – The number of simultaneous users will be less than 2000.

SLA – When the number of simultaneous users is less than 2000, agreed upon login times must be met. If they are not met or users under 2000 are not able to log in, the help desk will be notified and the problem will be addressed within one business day.

- Help desk training, procedures, staffing changes
- Timing and process of turnover to vendor support team
- Backup schedule
- Recovery procedures

Exercise 5.1 Develop a Test Case and Test Script

Develop test cases, scripts for the case study project. Also, update the Requirements Traceability Matrix and validate that all use cases will be tested.

Speedy Office Supplies Web Expansion Project Test Case/Script

Test Case ID TC001	Test the “Place Order” Use Case – UC001	Date:
Test Items: 1. Use Case UC001 2. Requirements document 3. Requirements Traceability Matrix		

Input specifications:	1. Customer file 2. Order file 3. Product file 4. Shipping file
Output specifications:	1. Order file 2. Product file 3. Credit Card file 4. Shipping file
Environmental needs:	Hardware: Legacy mainframe, Order Entry web server and data server Software: Inventory system, Order Fulfillment system, Shipping system, Order Entry system
Special procedural requirements:	Testing data will be backed up before the beginning of each testing session. All test files will be restored from backup if re-testing is requested or required.
Inter-case dependencies:	None

Procedure Steps	Expected Results	Pass/Fail
1. Customer selects “Order” option from web page.	2. System will display login dialog box	
3. Customer will enter account id and password	4. System will verify customer info and display order screen	
5. Customer will enter product codes and quantities	6. System will retrieve pricing and display order with totals	
7. Customer requests “Check-out”	8. System displays shipping info request screen	
9. Customer verifies shipping info	10. System accepts shipping and displays payment screen	
11. Customer pays for order	12. System accepts payment and displays confirmation	

Approvals:	Req’d	Signature and Date	
			Date:

Exercise 5.2 Using the Traceability Matrix to Validate Scope

Using the Requirements document, Use Cases, Test cases/scripts, and the Requirements Traceability Matrix from prior exercises, discuss below how you would use the Requirements Traceability Matrix to validate the requirements and scope of the Speedy Office Supplies Web Expansion Project.

The Requirements document and the Requirements Traceability Matrix are used to formalize the acceptance of the completed project deliverables. The Requirements Traceability Matrix also provides a thread through the project lifecycle that assists the project manager in tracing gathered requirements to work products and test cases. Now stakeholders can validate that all requirements have been developed, tested, and completed. The uses of these documents allows for a higher probability of success in meeting project objectives and stakeholder expectations.