

Requirements Validation and Functional Decomposition

Lesson 5: Requirement
Management

Lesson Objectives

- To understand the following topics:
 - What is Requirements Management?
 - Why do requirements change?
 - Stable and Volatile Requirements
 - Requirements Classification
 - Baseling Requirements
 - Requirements Traceability
 - Types of Requirements Traceability
 - Requirement Traceability Matrix
 - Maintaining Requirement Traceability
 - Requirement Traceability Matrix – Simple Example



Lesson Objectives

- To understand the following topics:
 - Change Management
 - Change Request Management
 - Change Management Process
 - Define Requirement Creep
 - Why does Requirement Creep occur?
 - Measures to control Requirement Creep
 - Requirement Metrics
 - Summary
 - Review Questions



What is Requirements Management?

- Requirement Management is a systematic process of drawing, establishing & documenting the software requirements of the system and maintaining agreement between the customer and the project team on changes to those requirements
- Requirements are inevitably incomplete and inconsistent
 - New requirements emerge during the process as business needs change and a better understanding of the system is developed
 - Different viewpoints have different requirements and these are often contradictory
- Requirement Management involves recognizing and anticipating that changes occur and to plan for them during the execution of the project
- Managing requirements is an on-going process that takes place throughout the project lifecycle



Copyright © Capgemini 2015. All Rights Reserved 4

Requirement Management – An Overview

One of the major reason behind the software failures is the way the project team acquire, document, agree upon and change the product requirements. The classic problem areas include informal information gathering, implied functionality, inadequately defined requirements, and a casual change process

The purpose of **Requirements management** is to manage the requirements of a project and to identify inconsistencies between those requirements and the project's plans and work products. Requirements management practices include change management and traceability.

The effective requirement management process involves maintaining a clear declaration of the requirements along with appropriate attributes & traceability. Managing changing requirements throughout the software development life cycle is the key to developing a successful solution.

The effective requirement management process leads to a successful software development, one that meets customer's needs and is developed on time and within the budget.

Why do requirements change?

- Change in software development: as inevitable as difficult to control!

"Stable requirements are the holy grail of software development."
(McConnell, 1993)



DILBERT © United Feature Syndicate, Inc.
Redistribution in whole or in part prohibited.



Copyright © Capgemini 2015. All Rights Reserved 5

Why do requirements change?

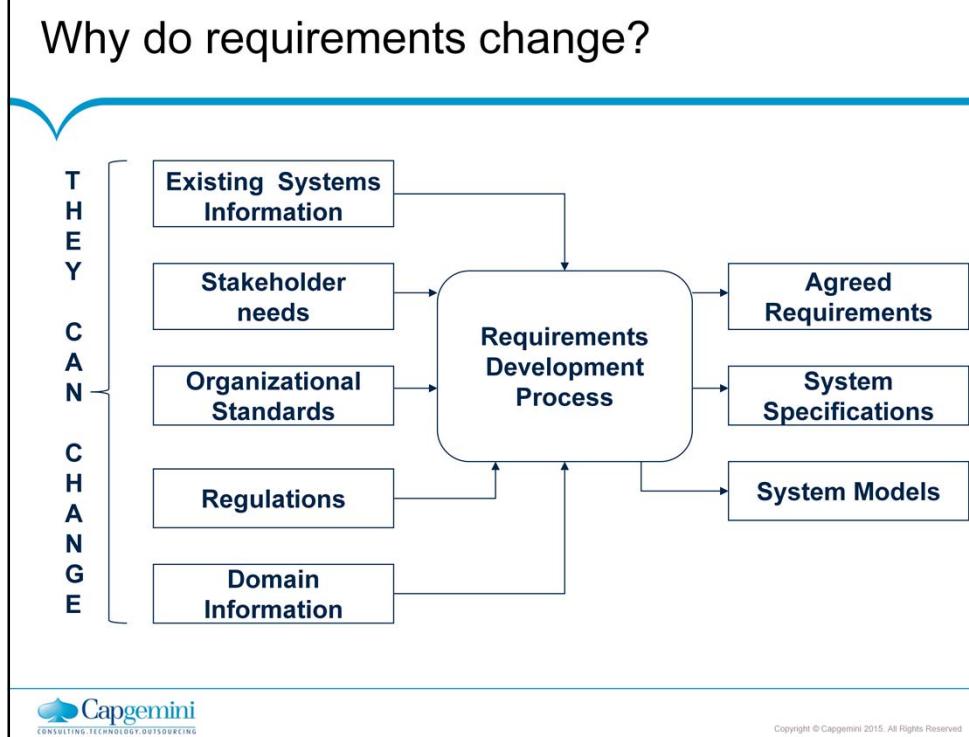
- Requirements change because :
 - Initial elicitation activities are imperfect
 - The priority of requirements from different viewpoints changes during the development process
 - Business needs evolve
 - Customers may specify requirements from a business perspective that conflict with end-user requirements
 - Customer's expectations change once they see the product taking shape
 - The business and technical environment of the system changes during its development



Copyright © Capgemini 2015. All Rights Reserved 6

Why do Requirements change?

1. Our understanding of the problem improved.
2. The problem being solved changed.
3. We failed to ask the right people the right questions at the right time.
4. We failed to create or follow a process to help manage change.
5. The users changed their minds or their perceptions.
6. The external environment changed.



Stable and Volatile Requirements

▪ Stable Requirements

- They are related to the core activities of the system and its domain
- For example, in an organization there will be requirements concerned with employees, departments, payroll etc.

▪ Volatile Requirements

- These are requirements that are likely to change during the system development process or after the system has become operational
- Examples of volatile requirements are requirements resulting from organization's leave policies or Income Tax policies enforced by the country's government bodies



Copyright © Capgemini 2015. All Rights Reserved 8

Requirements Classification

- **Mutable Requirements**

- These requirements change due to change in the environment in which the organization is operating.

- **Emergent Requirements**

- These requirements emerge as the system is designed and implemented.
 - They emerge as the customer's understanding of the system develops.

- **Consequential Requirements**

- Requirements that result from the introduction of the computer system.
 - This change may lead to the change in organization processes and open up new ways of working which may generate new system requirements.

- These requirements are affected by the way a system is to be used.

- **Compatibility Requirements**

- These requirements depend upon hardware, other systems or business processes within an organization.
 - An example could be an upgrade to an operating system, database software or a new version of hardware.



Copyright © Capgemini 2015. All Rights Reserved 9

Baselining Requirements

- The requirements are baselined at the end of the Requirements Development phase & ideally signed-off by the customer
- A requirements baseline is:
 - A snapshot in time of a set of requirements
 - Used as a mechanism to track changes as the project progresses
 - Constitutes agreement on scope between customer and development team
- Scope drives estimate, schedule, staffing, deadlines
- New baselines are typically created at major project milestones

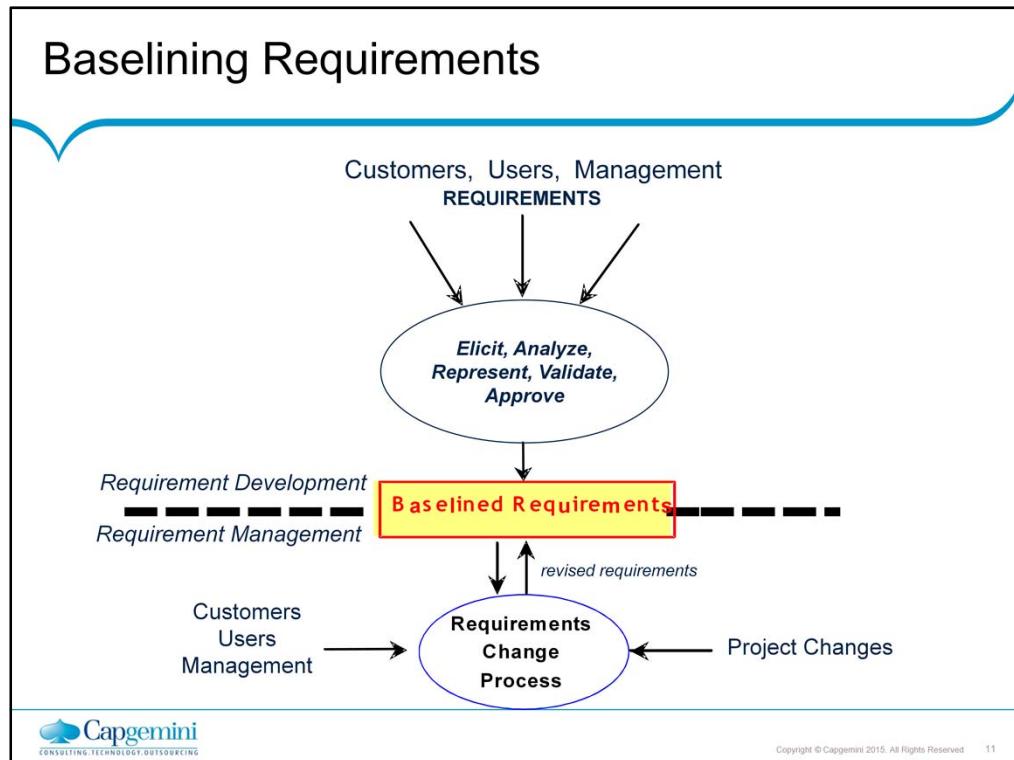


Copyright © Capgemini 2015. All Rights Reserved 10

Baselining Requirements

Usually projects start with unclear requirements and expectations. Lack of base lined requirements can result in chaos with lots of requirements changes resulting in requirements and scope creeps. Baselines can also help in acceptance testing and prototyping efforts. Baselines are especially valuable in fixed price contracts.

A baseline is all about getting to a common base agreement between stakeholders. It essentially involves setting the right expectations including responsibilities, risks, assumptions, deliverable and approaches. Once an agreement is reached; it could be put in source control to manage the base line going forward.



Baselining Requirements

Once sufficient requirements have been discovered and documented as an analyst you maybe required to help facilitate the planning of releases / versions.

Firstly you ensure that the requirements or features have been prioritized by the customer. This may require some diplomacy if you have a number of stakeholders with different priorities. This prioritization process is crucial and any conflicts must be resolved before starting development work where possible.

Secondly you will need to work with the development team to help them estimate the duration and risk of each requirement. Once completed you will be prepared to facilitate the discussion to start baseline planning.

The features that make up each version should be derived through a balance of priority value (determined by the project customer), effort required (determined by the development team) and perceived risk (determined by the development team). Baselined requirements are the start point for Requirements Management.

Requirements Traceability

- Requirements traceability is at the heart of the requirement management
- Requirements traceability refers to the ability to describe and follow the life of a requirement
- It is one of the essential activities of good requirements management
- Requirement traceability helps in assessing the impact of requirements change
- Traceability is used to track the relationship between each unique product-level requirement and its source

"The degree to which a relationship can be established between two or more products of the development process, especially products having a predecessor-successor or master-subordinate relationship to one another."

The IEEE Definition



Copyright © Capgemini 2015. All Rights Reserved 12

Requirement Traceability

Requirements traceability is concerned with documenting the life of a requirement. It should be possible to trace back to the origin of each requirement and every change made to the requirement should therefore be documented in order to achieve traceability. Even the use of the requirement after the implemented features have been deployed and used should be traceable.

Requirements come from different sources, like the business person ordering the product, the marketing manager and the actual user. These people all have different requirements on the product. Using requirements traceability an implemented feature can be traced back to the person, or group, that wanted it during the requirements elicitation. This can be used during the development process to prioritize the requirement, determining how valuable the requirement is to a specific user. It can also be used after the deployment when user studies show that a feature is not used, to see why it was required in the first place.

Types of Requirements Traceability

- Forward traceability

- This traceability is used to validate whether the project is evolving in the desired direction and for the right product
- It ensures that each requirement is implemented in the product and that each requirement is tested thoroughly
- This traceability maps requirements to test cases

- Backward or reverse traceability

- This traceability is used to validate that the product development is on the correct path of its development
- The objective behind this type of traceability is to ensure that the scope of the project is not expanding due to additional code, design elements or other tasks that are not specified in the requirements
- It maps test cases to requirements

- Bi-directional traceability

- This traceability ensures that all requirements are covered by test cases



Copyright © Capgemini 2015. All Rights Reserved 13

Requirement Traceability Matrix

- The classic way to perform traceability is by constructing a traceability matrix
- Traceability Matrix is a table containing requirements of a project and their relation to the engineering work products
- It ensures completeness in translating requirements to the delivered work products
- Advantages
 - Ensures completeness of testing against requirements
 - Facilitates the impact analysis of the requirements change on all the related work products
 - Enables scope analysis for regression testing
 - Helps judging requirements stability from a customer
 - Helps to analyze Requirements creep



Copyright © Capgemini 2015. All Rights Reserved 14

Requirement Traceability Matrix

- Matter of Fact
 - While traceability is one of the most difficult concepts to implement in Requirements Management, it is essential to accommodate change
- Suggested Tools
 - Rational Requisite Pro
 - Excel Sheet (Traceability template)



Copyright © Capgemini 2015. All Rights Reserved 15

Traceability looks very simple in theory, but is in fact very difficult to implement. Typically tools are used to manage requirements traceability.

Requirement Traceability Matrix

- Traceability in Excel

Config. Items	Event or Baseline Stage	Functional Specs	Design Specs	System Test Plan	Source
Requirements					
1	Baseline	Clause Rev 1.0	1.1/ Rev 1.0	Clause Rev 1.0	2.1/ Rev 1.0
	Change 1	Clause 1.1 /Rev 1.1	Clause 2.1 /Rev 1.1	Section 3.1 /Rev 1.1	x.c / func1
					y.c / func2

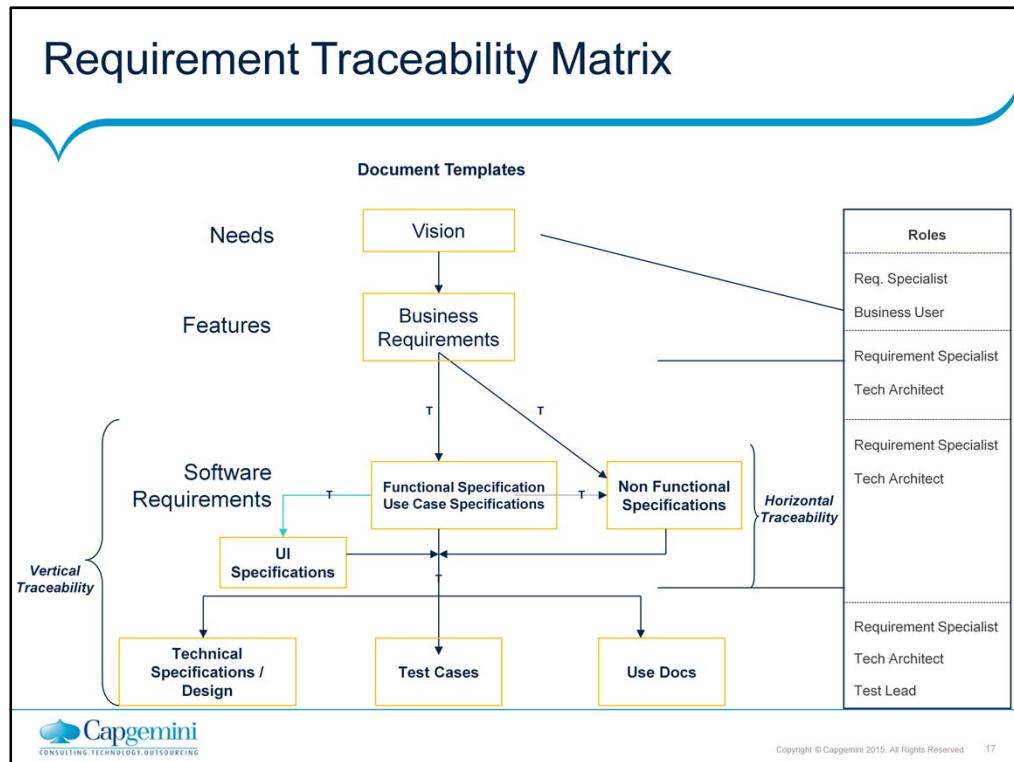

CONSULTING TECHNOLOGY OUTSOURCING

Copyright © Capgemini 2015. All Rights Reserved

16

Traceability as a general term is the "ability to chronologically interrelate the uniquely identifiable entities in a way that matters." The word *chronology* here reflects the use of the term in the context of tracking food from farm to shop, or drugs from factory to mouth. What matters in requirements management is not a *temporal* evolution so much as a *structural* evolution: a trace of where requirements are derived from, how they are satisfied, how they are tested, and what impact will result if they are changed.

The slide here depicts a typical way in which requirements can be traced using Excel.



The first step is to establish your requirements structure and the relationship of different requirements types to each other.

Based on this structure, you need to set up traceability links between all associated requirements or other project elements.

1. Trace top-level requirements to detailed requirements
2. Trace requirements to design
3. Trace requirements to test procedures
4. Trace requirements to user documentation plans

Requirement Traceability Matrix

Sr. No	Requirement Type	Traced to Requirement
1	Functional Requirement	Business Requirements
2	Non Functional Requirements	Business Requirements
3	Non Functional Requirements	Functional Requirement
4	Use Case Requirement	Business Requirements
5	UI Specifications	Use Case Requirement
6	Design Requirements	Functional Requirements
7	Design Requirements	Non-Functional Requirements
8	Design Requirements	Use Case Requirements



Copyright © Capgemini 2015. All Rights Reserved 18

Maintaining Requirement Traceability

- When should I start preparing the Traceability?
 - Should be prepared as soon as the high level functional requirements are available
 - Alternatively it can be done at the end of the Requirements Development phase when the requirements are sufficiently detailed
- When would I update the Traceability Matrix?
 - At the end of each life cycle phase that is tracked
 - When changes to the requirements are received (derived or explicit)
 - When changes to work products impact other work products



Copyright © Capgemini 2015. All Rights Reserved 19

Traceability needs to be on the radar right from the beginning. Ideally we should create the Traceability Matrix by the end of Requirements Development, and update it at the end of each subsequent phase and whenever there are changes in requirements.

Requirement Traceability Matrix – Simple Example

- Let's understand the concept of Requirement Traceability Matrix through a simple example of "FlyWithMe" Airways Web Application
- We will use the following business requirement & technical design requirement to understand the requirement traceability matrix
- Simple Business Requirement Document (BRD)

BR ID	Application/Module/Component Name	Roles Involved	Requirement Description
BR_1	Login Module	Customer	A customer need to login to the FlyWithMe Airways Website to avail the facility of Web Check-in

- Simple Technical Design Document (TDD)

TDR ID	Application/Module/Component Name	Requirement Description
TDR_16	Login Module	User ID should not be left blank
TDR_09	Login Module	Password should contain at least 1 alphanumeric value
TDR_29	Login Module	User should be able to login to the website upon entering valid User ID and Password



Copyright © Capgemini 2015. All Rights Reserved 20

Requirement Traceability Matrix – Simple Example

- Let's understand the concept of Requirement Traceability Matrix through a simple example of "FlyWithMe" Airways Web Application
- We will use the following business requirement & technical design requirement to understand the requirement traceability matrix
- Simple Business Requirement Document (BRD)

Test Case ID	Test Condition	Test Steps	Test Data	Expected Result
TC_1	To validate that user should be able to login to the airways website successfully with valid User ID and Password	Go to Login Page Enter User ID Enter Password Click on Login button	http://www.FlyWithMe.com/Login.aspx User1 fly@123	Login Successful

- Simple Technical Design Document (TDD)



Copyright © Capgemini 2015. All Rights Reserved 21

Requirement Traceability Matrix – Simple Example

- Step 2: Identify the Technical Requirement that this test case is verifying. For our test case, the technical requirement is TDR_29 is being verified.

Test Case ID	TDR ID	Test Condition	Test Steps	Test Data	Expected Result
TC_1	TDR_29	To validate that user should be able to login to the airways website successfully with valid User ID and Password	Go to Login Page Enter User ID Enter Password Click on Login button	http://www.FlyWithMe.com/Login.aspx User1 fly@123	Login Successful



Requirement Traceability Matrix – Simple Example

- Step 3: Identify the Business Requirement for which this TDR (Technical Requirement-TDR_29) is defined

Test Case ID	TDR ID	BR ID	Test Condition	Test Steps	Test Data	Expected Result
TC_1	TDR_29	BR_1	To validate that user should be able to login to the airways website successfully with valid User ID and Password	Go to Login Page Enter User ID Enter Password Click on Login button	http://www.FlyWithMe.com/Login.aspx User1 fly@123	Login Successful



Copyright © Capgemini 2015. All Rights Reserved 23

Requirement Traceability Matrix – Simple Example

- Step 4: We need to perform above process for all the test Cases. Later Extract the First 3 Columns from your Test Suite. RTM is Ready.

Test Case ID	TDR ID	BR ID
TC_1	TDR_29	BR_1



Copyright © Capgemini 2015. All Rights Reserved 24

Note - All the formats/templates used in RTM example are sample templates to maintain the simplicity of the example. They are not standard templates that are used in projects in IGATE.

Change Management

- The “Change Management” process is the sequence of steps or activities that a change management team or project leader would follow to apply change management to a project or change.
- A change management process consists of number of predefined processes and standards to be followed to manage changes to the system requirements.
- Managing requirement changes is an activity to identify, analyze, track, and report proposed changes and finally approve those changes to the product specification.
- The Change Control Board (CCB) that includes the project stakeholders are involved in the execution of change management process.

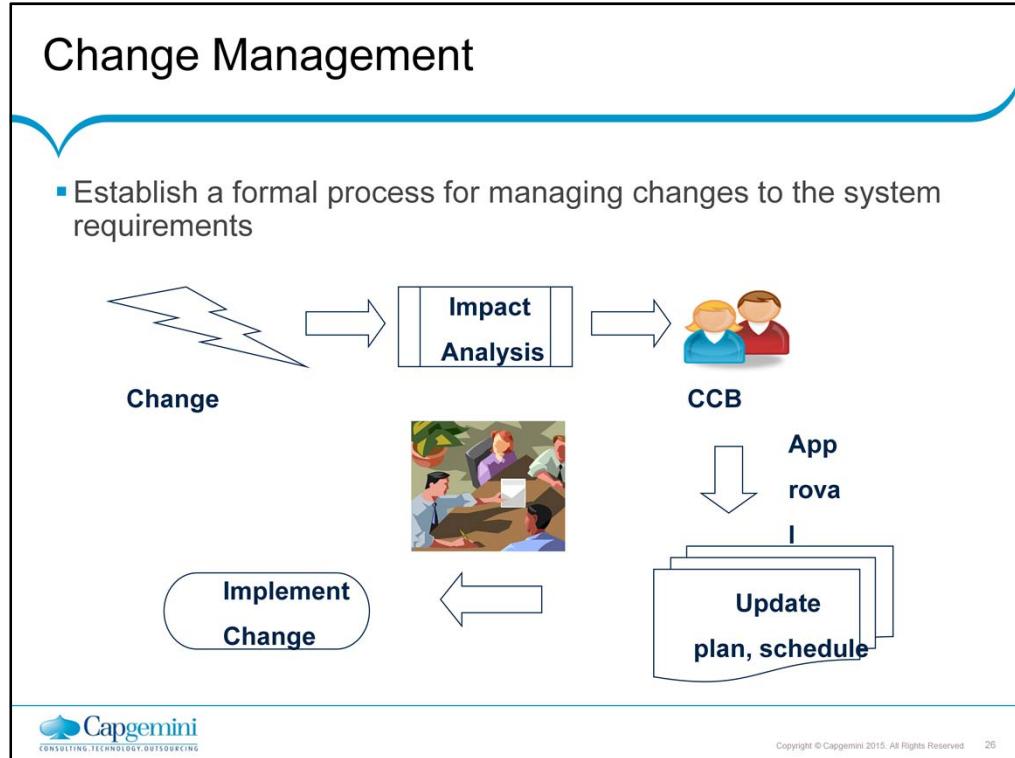


Copyright © Capgemini 2015. All Rights Reserved 25

Change Management

Managing requirement changes is an activity to identify, analyze, track, and report proposed changes and finally approve those changes to the product specification. As project develops, requirements may change or increase to accommodate changes in the project scope or design. When there is a request to add a new feature to the product, to enhance an existing specification due to a defect or failure or to adopt the changes in the various organization level or statutory and legal policies, a change request is created to modify the existing requirement specification. Those changes to the requirements can impact the project overall cost, resources allocated, and schedule planned for the delivery.

A change management process consists of number of predefined processes and standards to be followed to manage changes to the system requirements.



A typical change management process is illustrated here and explained further on the next slide. Key thing to remembers is that a structured process has to be followed before deciding to go ahead with the change.

Change Request Management

- Changes in Requirements can be due to
 - The customer requests a change to the existing requirements
 - The customer requests additional requirements
 - The project team identifies changes to the existing requirements
 - The project team identifies additional derived requirements
- Recording Changes (Change Logging Sheet)
 - Is used to maintain a consolidated record of all the changes (explicit or derived)
 - When a sizeable set of changes are received, the project may decide to raise a consolidated CR for approval of additional budget



Copyright © Capgemini 2015. All Rights Reserved 27

Change Management Process

- Identify potential change
 - Require new functionality
 - Encounter problem
 - Request change
- Do functional impact assessment upon any change request
- Analyze change request
 - Determine technical feasibility
 - Determine costs and benefits
- Evaluate change
- Obtain approval from customer on scope of change, impact & efforts needed
- Plan change
 - Analyze change impact
 - Create planning
- Implement change
 - Execute change
 - Propagate change
 - Test change
 - Update requirement artifacts with new requirement
 - Release change
- Review and close change
 - Verify change
 - Close change



Copyright © Capgemini 2015. All Rights Reserved 28

There are six main activities, which jointly form the change management process. They are: Identify potential change, Analyze change request, Evaluate change, Plan change, Implement change and Review and close change.

1. **Identify potential change:** The change in requirements is identified. This could come from an analysis of the requirements, new customer needs, or operational problems with the system. The change may be initiated by a customer/user, business analyst or any other stakeholder involved in defining the requirement.
2. **Analyze change request :** The proposed changes are analyzed. This includes how many other requirements and system components are affected by the change. The analysis are also carried out to understand approximate cost in terms of money, efforts & timer to successfully implement the change in the product and overall impact on the project schedule.
3. **Evaluate Change :** On the basis of change request analysis, the Change Control Board (CCB) decides whether to implement or not the change. This decision will be based on factors such as : the severity of the changes, cost versus benefit etc.
4. **Plan Change :** The first step in managing change is building awareness around the need for change and creating a desire among employees. If the CCB decides that the change is to be implemented, it is communicated to any stakeholders e.g. user/customer, project manager, project team etc. that are affected by that change. The change management team must develop a plan for successful implementation of the change. A key part of the change control process is to carry out an impact analysis of the proposed new or modified requirement. The impact analysis involves estimating the time, effort and cost of implementing the change and any other requirements that are affected by the change are also considered.

5. **Implement Change :** The change is implemented. The information about the new change implementation is then passed on to all of the people involved in the process and the ones affected by the change. It is also very important to verify that the new change that is introduced in the system is tested for its successful implementation. A set of amendments to the requirements document or a new document version is produced.
6. **Review and close changes :** The implementation team must determine the success of the change based on execution of the post implementation test plan and success criteria identified in the Change Request. If the change was not completed successfully as planned or is incomplete, the implementer must determine if the change should be reversed. The change will be closed off if the changes are successfully implemented. The benchmarks for a successful change are given below:
 - ❖ The change was implemented in accordance with the implementation plan
 - ❖ The change was implemented within the planned implementation timeframe
 - ❖ The change did not cause unplanned customer impact
 - ❖ The change met anticipated objectives defined in the Change Request
 - ❖ The change did not result in a system/application outage due to the execution of the back out plan

Define Requirement Creep

- Wikipedia defines Requirement Creep as given below:

"Scope creep (also called requirement creep and feature creep) in project management refers to uncontrolled changes or continuous growth in a project's scope. This can occur when the scope of a project is not properly defined, documented, or controlled."

- Requirement creep in project management refers to uncontrolled changes in the requirement during product development beyond those originally foreseen, leading to features that weren't originally planned and resulting risk to product quality or schedule



Copyright © Capgemini 2015. All Rights Reserved 30

Define Requirement Creep

- Even when there is a clearly defined project scope, one must be aware that Requirement creep can still occur during project development.
- Requirement creep can originate from several sources and is a leading cause of project failure when handled poorly.
- Requirement creep tends to occur when new features are added to product design that have already been approved without providing equivalent increase in the budget, time and resources.
- Additional requirements needed to achieve the new objectives can overwhelm the capacity of the resources allocated to the project resulting into project missing deadlines, budgets or complete failure.
- Therefore, preventing requirement creep and managing requirement creep is the key to successful project management.



Copyright © Capgemini 2015. All Rights Reserved 31

Why does Requirement Creep occur?

- In a Computerworld survey of 160 I.T. professionals, 80% reported requirements creep either “always” or “frequently”.
- There are a number of reasons for requirement creep, and the following are a few of the most common:
 - Poor requirement analysis
 - The customers are not sure enough about what they want from the system and they end up stating vague requirements
 - Not involving customers early enough
 - This refers to having false confidence that you know exactly what the customers expect from the system
 - It is very important to involve customers in the requirement analysis as well as design phase
 - Insufficient detailing on the complexity of the project
 - Many projects run into problems because they are executed for the first time and there is no enough detailed information available on what to expect from the project and how to implement the same in a standard manner



Copyright © Capgemini 2015. All Rights Reserved 32

Why does Requirement Creep occur?

- Lack of change control

- You can expect requirement creep in most projects, therefore it is important to design a process to manage these changes

- Gold Plating

- This term is given to the practice of exceeding the scope of a project in the belief that a value is being added
 - As a team proceeds through the various phases of a project, frequently one or more of the team members will strive to improve or perfect the product
 - These changes inevitably consume time and budget and are not guaranteed to increase customer satisfaction

- Unwillingness to say no to a client

- The project team's or an individual's desire to please the customer and reluctance to say "no" can also lead to requirement creep

- One little change won't matter

- There can be a situation when customers ask for a change, if proper and required impact analysis of the change is not performed then it can give a false impression of change being manageable which can fit into current schedule and cost



Copyright © Capgemini 2015. All Rights Reserved 33

Measures to control Requirement Creep

- The best way to minimize scope creep is to define the requirements up front as thoroughly as possible
- Utilize different techniques such as prototyping and joint application development (JAD) sessions, to thoroughly explore and define the business and technical requirements
- Following are some of the common measures those can be used to minimize requirement creep
 - Utilize various techniques for more thoroughly defining user requirements up front
 - Involve the customers in the earliest stages of the project possible
 - Achievable goals should be set
 - Prioritize requirements into must-haves versus nice-to-haves
 - Project managers have to learn when to say no and when to say yes
 - When the client wants to change or add a requirement, the change or addition should be analyzed for resource, cost, and schedule impacts
 - Perform constant internal review to make sure the project is on track and within scope
 - Set a timeline or due date for all tasks
 - Have a tracking system for tasks, due dates, and action items



Copyright © Capgemini 2015. All Rights Reserved 34

Requirement Metrics

Process	Sub Process	Objective	Metrics
Requirements Development	Elicitation	<ol style="list-style-type: none">Minimize the scope creepIncrease requirements stability	<ol style="list-style-type: none">Requirements Creep Index (RCI)Requirements Stability Index (RSI)
	Elicitation	<ol style="list-style-type: none">Increase requirements elicitation	<ol style="list-style-type: none">Elicitation Efficiency (EEI)
Requirements Management	Query Resolution	<ol style="list-style-type: none">Increase query resolutionDecrease time for closing a query	<ol style="list-style-type: none">Query Resolution Index (QRI)Average Query Turnaround Time (QTAT)



Copyright © Capgemini 2015. All Rights Reserved 35

Requirement Metrics

Metrics	Overview	Calculation
Requirements Creep Index (RCI)	A measure of scope creep over gathered requirements	$((\text{Effort due to scope creep}) * 100) / (\text{Approved efforts in person days as approved by customer})$
Requirements Stability Index (RSI)	A measure of frequency of change in baselined requirements over project lifecycle	$(N - C)/N$, where, N = Number of requirements identified and approved for the project. C = Number of suggested changes in requirement because of issues at any time in the project's phase.
Elicitation Efficiency (EEI)	A measure of the efficiency of elicitation process	Use of a template for calculating EEI
Query Resolution Index (QRI)		Total number of Queries Resolved / Total number of Queries
Average Query Turnaround Time (QTAT)	Time taken to resolve a functional query	$\Sigma(\text{End date} - \text{Start date}) \text{ of all queries} / \text{Total number of closed queries for the reporting period}$



Copyright © Capgemini 2015. All Rights Reserved 36

Summary

- In this lesson, you have learnt:
 - What is Requirements Management?
 - Why do requirements change?
 - Stable and Volatile Requirements
 - Requirements Classification
 - Baseline Requirements
 - Requirements Traceability
 - Requirement Traceability Matrix
 - Maintaining Requirement Traceability
 - Change Management
 - Requirement Creep



Review Question

- Question 1: Volatile Requirements are likely to change during the system development process or after the system has been become operational.
 - True/ False
- Question 2: The requirements are baselined at the beginning of the Requirements Development phase & ideally signed-off by the development team.
 - True/ False
- Question 3: _____ is a measure of frequency of change in baselined requirements over project lifecycle.
- Question 4: _____ requirements are the requirements that are depend upon hardware, other systems or business processes within an organization.



Copyright © Capgemini 2015. All Rights Reserved 38

Review Question

- Question 5: Which of the following requirement traceability is used to validate whether the project is evolving in the desired direction and for the right product?
 - Option 1: Forward traceability
 - Option 2: Backward traceability
 - Option 3: Bi-directional traceability
 - Option 4: All of the above

