

# Requirements Validation And Functional Decomposition

Lesson 00:

IGATE is now a part of Capgemini

People matter, results count.



©2016 Capgemini. All rights reserved.  
The information contained in this document is proprietary and confidential. For Capgemini only.

## Document History

Date	Course Version No.	Software Version No.	Developer / SME	Change Record Remarks
	0.1D	NA		Content Creation
	0.1	NA		Review
May-2009	1.0	NA	Priya Rane	Material Revamp
June-2011	1.1	NA	Selva Lakshmi P.	Material Revamp
Aug-2013	2.0	MS Excel 2003	Shilpa Bhosle	Material Revamp
April – 2015	2.1	MS Excel 2010	Shilpa Bhosle	Material Revamp



Copyright © Capgemini 2015. All Rights Reserved 2

## Course Goals and Non Goals

### ■ Course Goals

- At the end of this course participants gain an understanding of what is requirement and what is requirement engineering
- Differentiate between two types of requirements – Functional Requirement and Non Functional requirements
- What is the importance of requirements in projects?
- What are the characteristics of good requirements?
- What is Requirement Itemization?
- How to use different templates to perform requirement itemization with the help of a case study?
- How to manage and successfully execute change in the requirement during Project life cycle?
- What is Requirement Traceability?

### ■ Course Non Goals

- This course does not cover tools training to manage requirements



Copyright © Capgemini 2015. All Rights Reserved 3

## Pre-requisites

- Fundamental knowledge of Software Testing
- Different types of Software Testing techniques
- Different types of Software Testing



Copyright © Capgemini 2015. All Rights Reserved 4

## Intended Audience

- Test Engineers and Senior Test Engineers



## Day Wise Schedule

- Day 1

- Lesson 1: Introduction to Requirements Engineering
- Lesson 2: Evolution and Types of Requirements
- Lesson 3: Requirements Itemization

- Day 2

- Lesson 3: Requirements Itemization (Cont.)
- Lesson 4: Requirements Gathering – Challenges & Techniques
- Lesson 5: Requirement Management



Copyright © Capgemini 2015. All Rights Reserved 6

## Table of Contents

- Lesson 1: Introduction to Requirements Engineering
  - 1.1 Requirements Engineering and Projects failure
  - 1.2. Why do Projects fail? – Current Survey
  - 1.3 Requirements Contains Defects
  - 1.4 Importance of Requirements – Some statistics!
  - 1.5 Why do we need good requirements?
  - 1.6 What is a Requirement?
  - 1.7 Requirement Definition
  - 1.8 Why are Requirements important?
  - 1.9 Requirements Engineering



Copyright © Capgemini 2015. All Rights Reserved 7

## Table of Contents

- Lesson 2: Evolution and Type of Requirements
  - 2.1 Evolution of Requirements
  - 2.2 Who provides the Requirements?
  - 2.3 Types of Requirements
  - 2.4 Functional Vs Non-Functional Requirements
  - 2.5 Do not overlook the “Non-Functional Requirements”!
  - 2.6 Non Functional Requirements: FURPS +
  - 2.7 Other Non Functional Requirements: “+”
  - 2.8 What is a good software requirement?



Copyright © Capgemini 2015. All Rights Reserved 8

## Table of Contents

- Lesson 3: Requirements Itemization
  - 3.1 Requirements Itemization - How?
  - 3.2 Application Background
  - 3.3 Assumptions
  - 3.4 Explicit requirements
  - 3.5 Implicit requirements
  - 3.6 Interface Requirements
  - 3.7 Requirements Analysis to Test Scenarios
  - 3.8 Testable Items
  - 3.9 Non Testable Items
  - 3.10 Error Conditions
  - 3.11 Application Invocation
  - 3.12 Application Termination



Copyright © Capgemini 2015. All Rights Reserved 9

## Table of Contents

- Lesson 3: Requirements Itemization (Cont.)
  - 3.13 File Handling
  - 3.14 Requirement Prioritization
  - 3.15 Case Study – Equipment Tracking System
  - 3.16 Template – 1 Equipment tracking System
  - 3.17 Template – 2 Equipment tracking System



Copyright © Capgemini 2015. All Rights Reserved 10

## Table of Contents

- Lesson 4: Requirements Gathering – Challenges & Techniques
  - 4.1 Requirements Gathering – A Typical Illustration
  - 4.2 Requirement Gathering Patterns
  - 4.3 Challenges in Requirements
  - 4.4 Challenges – Clarity of requirements
  - 4.5 Challenges – Communication
  - 4.6 Ambiguity From a Requirements Perspective - Pitfalls of the English language
  - 4.7 Ambiguity and Pitfalls
  - 4.8 Ambiguity Checklist
  - 4.9 Ambiguity Review
  - 4.10 Requirement Gathering - Skills Required
  - 4.11 Tips to Requirement Gathering
  - 4.12 Identification and Verification of Requirements



Copyright © Capgemini 2015. All Rights Reserved 11

## Table of Contents

- Lesson 5: Requirement Management
  - 5.1 What is Requirements Management?
  - 5.2 Why do requirements change?
  - 5.3 Stable and Volatile Requirements
  - 5.4 Requirements Classification
  - 5.5 Baseling Requirements
  - 5.6 Requirements Traceability
  - 5.7 Types of Requirements Traceability
  - 5.8 Requirement Traceability Matrix
  - 5.9 Maintaining Requirement Traceability
  - 5.10 Requirement Traceability Matrix – Simple Example



Copyright © Capgemini 2015. All Rights Reserved 12

## Table of Contents

- Lesson 5: Requirement Management
  - 5.11 Change Management
  - 5.12 Change Request Management
  - 5.13 Change Management Process
  - 5.14 Define Requirement Creep
  - 5.15 Why does Requirement Creep occur?
  - 5.16 Measures to control Requirement Creep
  - 5.17 Requirement Metrics



Copyright © Capgemini 2015. All Rights Reserved 13

## References

- Student material:
  - Class Book (presentation slides with notes)



Copyright © Capgemini 2015. All Rights Reserved 14

## Next Step Courses (if applicable)

- None



Copyright © Capgemini 2015. All Rights Reserved 15

## Other Parallel Technology Areas

- NA



Copyright © Capgemini 2015. All Rights Reserved 16

# Requirements Validation and Functional Decomposition

Lesson 1: Introduction to Requirements Engineering

## Lesson Objectives

- To understand the following topics:
  - Requirements Engineering and Projects failure
  - Why do Projects fail? – Current Survey
  - Requirements Contains Defects
  - Importance of Requirements – Some statistics!
  - Why do we need good requirements?
  - What is a Requirement?
  - Requirement Definition
  - Why are Requirements important?
  - Requirements Engineering
  - Summary
  - Review Questions



## Requirements Engineering and Projects failure

- Requirements Engineering is one of the challenging and key tasks in the development of software products
- It is one of the key reasons found in surveys, that causes the failure of software projects due to poor requirement management activity
- We should understand the problem before we can express the requirements for a correct solution
- Otherwise, we will develop the software that fails to provide customer satisfaction and remains on the shelf rather than off
- Expressing a set of complete, consistent, and correct requirements is conceptually complex, but essential activity in the quest to develop high-quality & useful software
- According to the Standish Group's 1995 CHAOS survey, the top two "project impaired" factors were incomplete requirements and lack of user involvement



Copyright © Capgemini 2015. All Rights Reserved 3

## Requirements Engineering and Projects failure

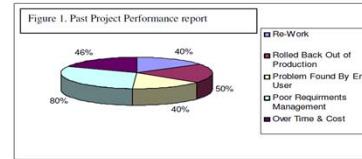
- Requirement problems are the single No.1 reason for project to fail - (Donald Firesmith, SEI, Carnegie Mellon University 2003)
  - Bad requirements cause failures
  - Extensively over budget
  - Extensively past schedule
  - Extensively reduced scope
  - Poor quality applications
  - Not considerably used when delivered
  - Sometimes getting cancelled
- Reworking requirements cost 40-50% of project effort (Caper Jones)
- Percentage of defects originating from requirements is estimated as 50-60% (Karl Weigers 2001)



Copyright © Capgemini 2015. All Rights Reserved 4

## Why do Projects fail? – Current Survey

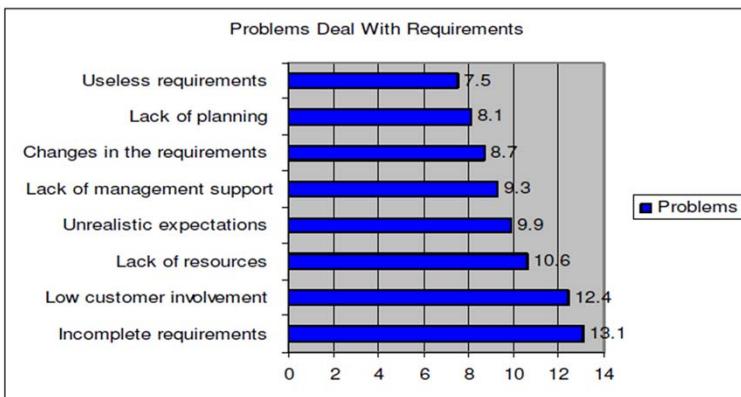
- CIO Magazine
  - Analysts report said 71% of software projects that fail due to poor requirements
  - management, making it the single biggest reason for project failure. [16]
- Standish Report
  - The Standish CHAOS Report, which surveyed 9,236 IT projects, found that the top three causes of project failure were lack of user input, incomplete requirements or changing requirements
  - According to the Standish Group International CHAOS Survey – U.S.A past project performance report is shown in following chart:
  - International Journal of Software Engineering & Applications (IJSEA)
  - Vol.2, No.4, October 2011



Copyright © Capgemini 2015. All Rights Reserved 5

## Why do Projects fail? – Current Survey

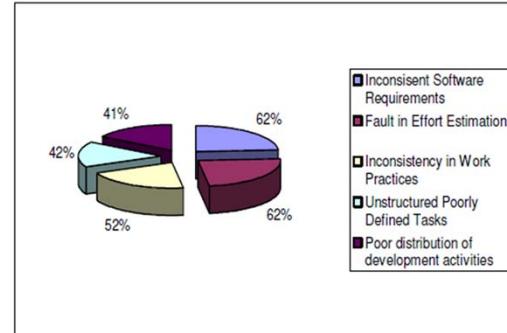
- According to the Standish Group International CHAOS Survey – U.S.A the Factors for Project Failure Deal with Requirements is shown in following chart:



Copyright © Capgemini 2015. All Rights Reserved 6

## Why do Projects fail? – Current Survey

- Survey of European Software Organizations
  - A recent survey of European software organizations identified that more than 40% perceived that they had major problems in managing customer requirements
- International Journal of Software Engineering & Applications (IJSEA),
  - Vol.2, No.4, October 2011



## Requirements Contains Defects

- The percentage of defects originating during requirement engineering are estimated as :
  - 50% - Karl Wiegers 2001
  - 42% - A. Winnegrove
  - 60%-64% - requirements and design EBG Consulting
- Characteristics of defective requirements:
  - Lack of Cohesiveness
  - Lack of Completeness
  - Lack of Correctness
  - Lack of Consistency
  - Lack of Project Relevance
  - Lack of Testability, Usability, Validatability
  - Ambiguous



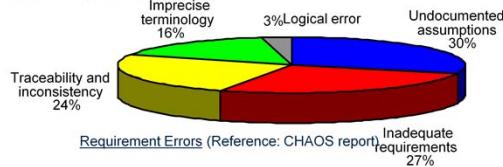
Copyright © Capgemini 2015. All Rights Reserved 8

## Importance of Requirements – Some statistics!

No.	Cause of project failure	%
1	Incomplete requirements	13.1
2	Lack of User involvement	12.4
3	Lack of resources	10.6
4	Unrealistic expectations	9.9
5	Lack of executive support	9.3
6	Changing requirements	8.7
7	Lack of Planning	8.1
8	Didn't need it any more	7.5
9	Lack of IT management	6.2
10	Technology illiteracy	4.3
	Others	9.9

Over 40% of root causes are related to user requirements

\* "Experiences using Formal Methods for Requirements Modeling." Easterbrook, et al.



The cost to fix a Requirements Defect after deployment can be over 200 times!

Relative cost to repair \*

\* Alan Davis (Software requirements: Objects, Functions and States)



Copyright © Capgemini 2015. All Rights Reserved. 9

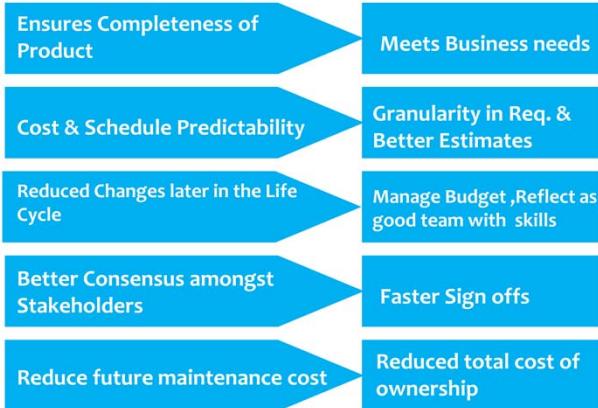
The most expensive errors to correct are those introduced at Requirements Time and not found until after release to the customer. These same errors are also the least expensive to correct if found during Requirements Time. The longer they exist, the more expensive they become.

The figure shows the order-of-magnitude relative costs of fixing a requirement error during requirements capture, coding, and testing.

Note: This formula is based on a waterfall process model. While an iterative process should reduce the cost of requirement errors, it still holds true that the later an error is found in the process, the more expensive it is to fix.

## Why do we need good requirements?

- Why do we need the requirement phase?
- What is it supposed to achieve?



## What is a Requirement?

- Requirement is a descriptions of how a product or service should act, appear or perform
- Requirements can range from high-level abstract statements of services or system constraints to detailed functional specifications
- Requirements are discovered from the customer outlining in terms of what they wish the system must do, and mentioned in a language the customer understands
- A requirements document then becomes the contract between the customer and the software developers on the product that will be delivered
- The Institute of Electrical and Electronics Engineers (IEEE) defines a requirement as :
  - a condition or capability needed by a user to solve a problem or achieve an objective
  - a condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed document
  - a documented representation of a condition or capability as in definition 1 or 2 [IEEE 1990a]



Copyright © Capgemini 2015. All Rights Reserved 11

## Requirement Definition

- In a simple term, Requirement is a capability requested by a user to solve a problem or to get something done
- It is a specification that mentions what is required from the system
- A Requirement is a single, measurable objective that a system must satisfy
- Where does it fit in the big picture of software development life cycle (SDLC) ?



Copyright © Capgemini 2015. All Rights Reserved 12

## Why are Requirements important?

- The following quote from Fredrick Brooks illustrates why requirements are so important:

*"The hardest part of building a software system is deciding precisely what to build. No other part of the conceptual work is as difficult as establishing the detailed technical requirements, including all of the interfaces to people, to machines, and to other software systems. No other part of the work so cripples the resulting system if done wrong. No other part is more difficult to rectify later"*
- One can end up doing a perfect job of building the wrong product if the clear and unambiguous system requirements are not captured

*"If you don't get the requirements right, it doesn't matter how well you do anything else."*

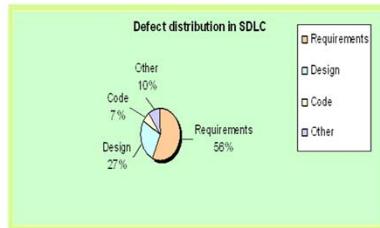
Karl Wiegers (2004)



Copyright © Capgemini 2015. All Rights Reserved 13

## Why are Requirements important?

- Numerous surveys and studies indicate that one of the major challenges in IT system development is the determination of software requirements
- System testing is about comparing the actual behavior of the software, with the specified behavior as described by the requirements
- Since the requirements are used as a base, the quality of those influences the ability to perform system testing activity efficiently and effectively



Copyright © Capgemini 2015. All Rights Reserved 14

### Why are Requirements important?

Numerous surveys and studies indicate that one of the major challenges in IT system development is the determination of system requirements. It might be surprising that after several decades of industry experience in these endeavours, developers would still pursue projects for which the requirements are not abundantly clear. In addition to direct financial costs, failed projects can also lead to expensive litigation. Without well-defined managers can not plan a project, developers will not know what to build, customers will not know what to expect from the system, and there will not be any way to validate that the system satisfies the need of the customers.

# Requirements Engineering

## ■ Definition of Requirement Engineering

- “Requirements engineering is the branch of software engineering concerned with the real-world goals, functions and constraints on software systems”
- Software requirements must be clear, correct, unambiguous, specific, and verifiable
- Requirement Engineering helps to get the precise information on:
  - Requirements (Analyzing)
  - What stakeholders need (Validating)
  - What designers have to build (Defining)
  - They have done so correctly (Verifying)
  - Capable to cope with the changes (Evolution)



Copyright © Capgemini 2015. All Rights Reserved 15

## What is Requirement Engineering?

Requirements Engineering is the process of establishing the services that the customer requires from the system and the constraints under which it is to be developed and operated. We first elicit requirements from the sources that are available to us like experts, stakeholders, customer or the current software and then model them to specify a solution. The elicitation & modelling requirements are two interrelated processes.

The gradual stabilization of such models in terms of the requirements leads to a satisfactory candidate specification, which then must be validated and verified. This gives stakeholders feedback on the interpretation of their requirements so they can correct misunderstandings. Requirements engineering denotes both the process of specifying requirements by studying stakeholder needs and the process of systematically analyzing and refining those specifications.

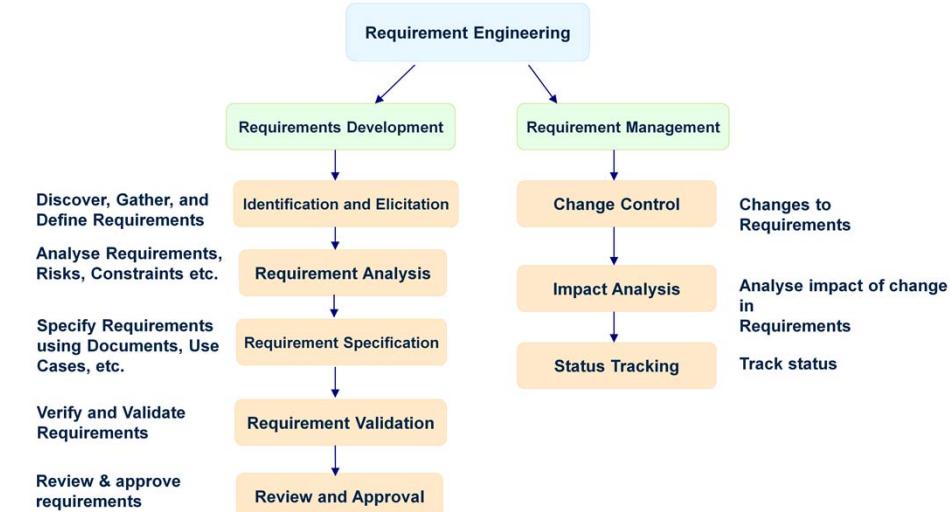
## Requirements Engineering

- Requirements Engineering is a disciplined, process-oriented approach to the definition, documentation, and maintenance of software requirements throughout the software development life cycle
- Software requirements engineering is made up of two major processes: “Requirements Development” and “Requirements Management”
  - Requirements development involves all of the activities that are part of eliciting, analyzing, specifying, and validating the requirements
  - Requirements management involves the activities that are part of requesting changes to the baselined requirements, performing impact analysis for the requested changes, approving or disapproving those changes, and implementing the approved changes



Copyright © Capgemini 2015. All Rights Reserved 16

# Requirements Engineering



## Requirement Engineering Process :

Software requirements engineering is made up of two major processes: requirements development and requirements management.

**Requirement Development :** Requirements development involves all of the activities that are part of eliciting, analyzing, specifying, and validating the requirements

**Requirement Elicitation:** The requirements elicitation stage includes all of the activities involved in identifying the requirements. There are different techniques available those can be used to elicit requirements which includes stakeholder interviews, focus groups, facilitated requirements workshops, observations of current work processes, questionnaires and surveys, analysis of competitor's products, and benchmarking of industry practices. Elicitation also includes those activities that explore how software can meet organizational goals, what alternatives might exist, and how they affect various stakeholders.

**Requirement Analysis:** In this stage, the stakeholder's needs, assumptions, and other information identified during requirements elicitation are combined together and refined into further levels of detail. This step includes representing the requirements in various forms including prototypes and models, establishing priorities, analyzing feasibility, and looking for gaps that identify missing requirements.

**Requirement Specification:** The stage in which the requirements are formally documented is called Requirement Specification.



The requirement specification can take one of many forms.

In case of small projects requirements information can be documented in a single Software Requirement Specification (SRS) document.

In case of larger projects, Business requirements may be documented in a business requirements document (BRD)

The software functional and non-functional requirements and the constraints may be documented in an SRS

External interfaces may be included in the SRS or in separate external interface requirements document

**Requirement Validation :** In this stage we validate the requirements to ensure that they are well written, complete, and will satisfy the customer needs. This stage may result into revisiting the other stages in the requirements development process if found defects, gaps in the requirements. One of the primary tools for requirements validation is to conduct formal peer reviews of the requirements specification documents before they are baselined.

**Review & Approval:** After one or more iterations through the software requirements development process, part or all of the requirements are deemed “good enough” to baseline and become the basis for software design and development. These requirements need to undergo a proper review and approval phase wherein various stakeholders perform the systematic review of these requirements and obtain approvals on the same.



Copyright © Capgemini 2015. All Rights Reserved 19

**Requirement Management:** Requirements management covers the activities that are involved in requesting changes to the baselined requirements, performing impact analysis for the requested changes, approving or disapproving those changes, and implementing the approved changes. Requirements management also includes the activities used for ensuring that work products and project plans are kept consistent and tracking the status of the requirements as one progresses through the software development process.

## Summary

- In this lesson, you have learnt:
  - Importance of Requirement Engineering and project failures
  - Why do we need good requirements?
  - What is a Requirement?
  - Requirement Definition
  - Requirement Engineering



Summary



Copyright © Capgemini 2015. All Rights Reserved 20

## Review Question

- Question 1: In which of the following stages of Requirement Engineering the activities related to identifying the requirements are carried out?
  - Option 1: Requirement Elicitation - Correct
  - Option 2: Requirement Analysis
  - Option 3: Requirement Specification
  - Option 4: Requirement Validation
- Question 2: In which stage of Requirement Engineering we do manage changes in requirement?
  - Option 1: Requirement Management
  - Option 2: Requirement Validation
  - Option 3: Change Control Management
  - Option 4: Requirement Analysis



Copyright © Capgemini 2015. All Rights Reserved 21

## Review Question

- Question 3: Requirements are discovered from the customer outlining in terms of what they wish the system must do, and mentioned in a language the customer understands.
  - True/ False
- Question 4: \_\_\_\_\_ involves all of the activities that are part of eliciting, analyzing, specifying, and validating the requirements.
- Question 5: In \_\_\_\_\_ stage we validate the requirements to ensure that they are well written, complete, and will satisfy the customer needs.



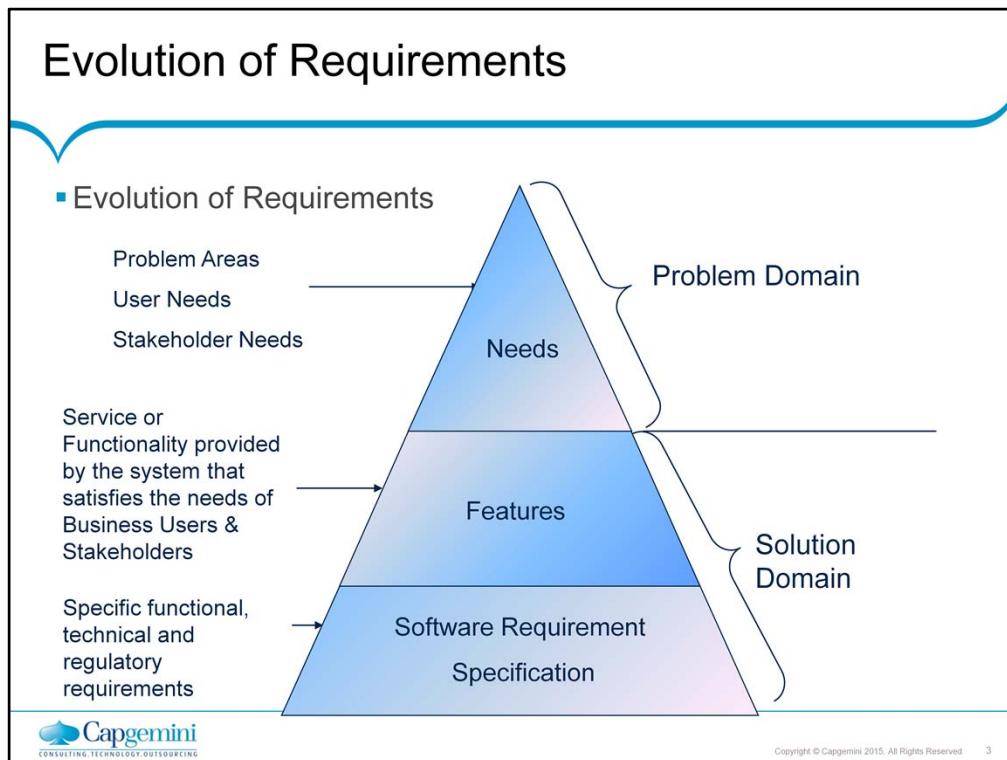
## **Requirements Validation and functional decomposition**

Lesson 2: Evolution and Types of Requirements

## Lesson Objectives

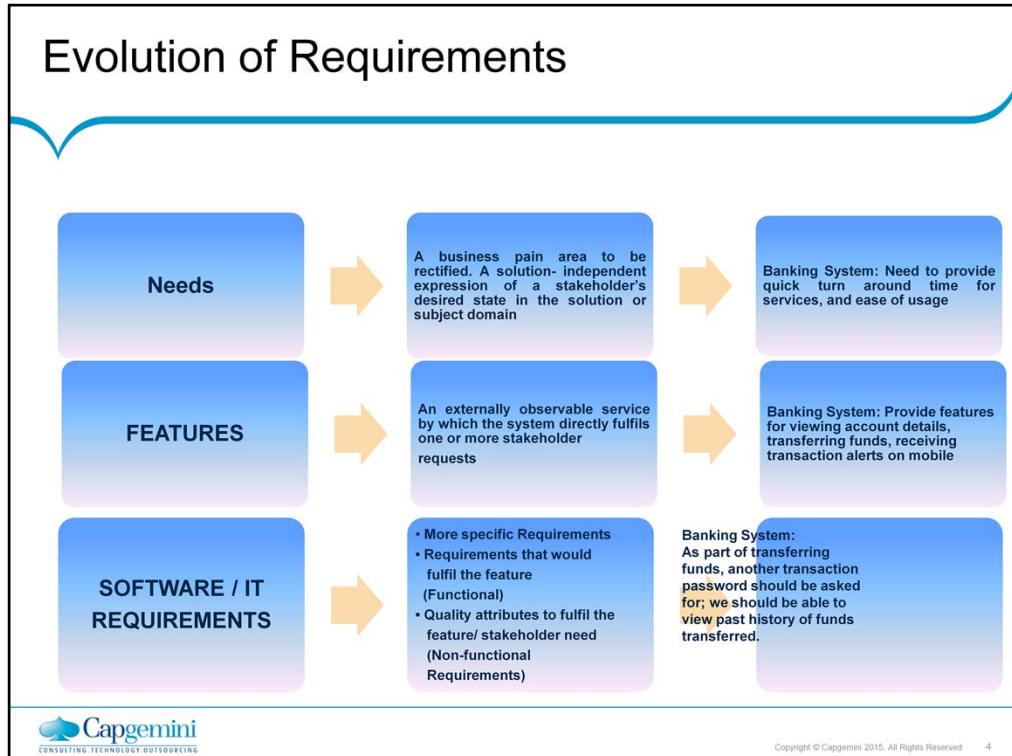
- To understand the following topics:
  - Evolution of Requirements
  - Who provides the Requirements?
  - Types of Requirements
  - Functional Vs Non-Functional Requirements
  - Do not overlook the “Non-Functional Requirements”!
  - Non Functional Requirements: FURPS +
  - Other Non Functional Requirements: “+”
  - What is a good software requirement?
  - Summary
  - Review Questions





### Evolution of Requirements:

Evolution involves the translation of stakeholder requests into a set of system features. These in turn are detailed into specifications for functional and nonfunctional requirements. Detailed specifications are translated into test procedures, design, and user documentation.



## Who provides the Requirements?

- The “Stakeholders” are individuals who affect or are affected by the proposed software product
- The requirements engineering process provides the best opportunity to consider the various stakeholders and their requirements from the software product
- Following are the different stakeholders :
  - Customers – These are the entities that request purchase, and/or pay for the software product in order to meet their business objectives
  - End Users – They actually use the product directly or use the product indirectly by receiving reports, outputs, or other information generated by the product
  - Development Team – They include individuals and teams that are part of the organization that develops the software product.
    - Business Analyst - Responsible for eliciting the requirements from the customers, users, and other stakeholders, analyzing the requirements, writing the requirements specification, and communicating the requirements to development and other stakeholders
    - Designers –They are responsible for translating the requirements into the software's architectural and detailed designs that specify how the software will be implemented



Copyright © Capgemini 2015. All Rights Reserved 5

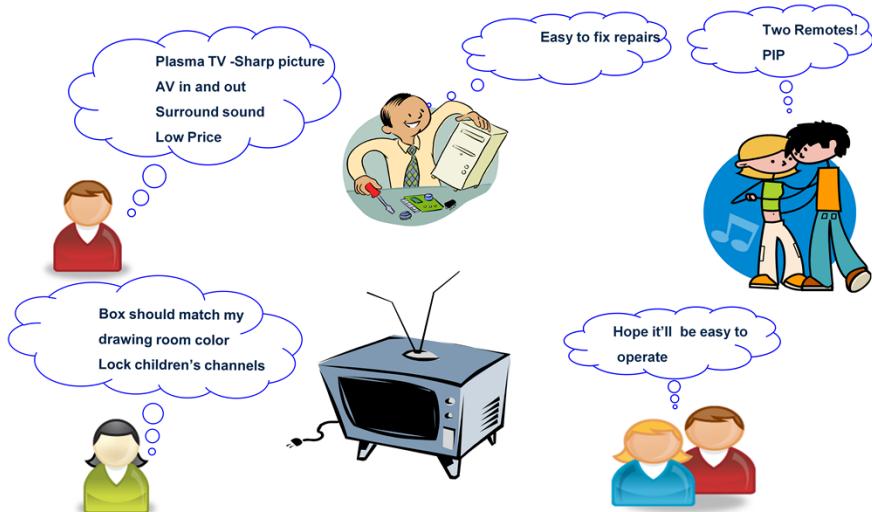
## Who provides the Requirements?

- Business Analyst - Responsible for eliciting the requirements from the customers, users, and other stakeholders, analyzing the requirements, writing the requirements specification, and communicating the requirements to development and other stakeholders
- Designers –They are responsible for translating the requirements into the software's architectural and detailed designs that specify how the software will be implemented
- Developers - The developers are responsible for implementing the designs by creating the software products
- Testers - The testers use the requirements as a basis for designing test cases that they use to execute the software under specific, known conditions to detect defects and provide confidence that the software performs as specified
- Other Stakeholders: There may be other stakeholders interested in the requirements too.
- Following is the list of other requirements stakeholders include:
  - Legal or contract management
  - Manufacturing or product release management
  - Sales and marketing
  - Upper management
  - Government or regulatory agencies



Copyright © Capgemini 2015. All Rights Reserved 6

## Different stakeholders....So different requirements



## Types of Requirements

- Requirements are typically placed into two categories
  - Functional Requirements, which say what the system should do or how it should work
  - Functional requirements define how software behaves to meet user needs
  - Consider an example of a health insurance company designing a claims system
  - Below are some of the functional requirements that the system will include
    - Determine Claimant Eligibility
    - Pay Claim
    - Calculate Premium
  - Non-Functional Requirements, which say what constraints there are on the system and its development
  - They represent quality attributes of the system
  - The quality attributes include:
    - Availability
    - Maintainability
    - Performance
    - Portability
    - Reliability
    - Robustness
    - Security
    - Scalability etc.



Copyright © Capgemini 2015. All Rights Reserved 8

### Types of Requirements

**Functional Requirements** - Functional requirements capture the intended behaviour of the system. Functional requirements explain what has to be done by identifying the necessary task, action or activity that must be accomplished. It specifies actions that a system must be able to perform. Functional requirements thus specify the input and output behaviour of a systems.

**Non Functional Requirements** - Non functional requirements are the requirements that specify the criteria that can be used to determine the operation of a system, rather than a specific behaviour. Non functional Requirements specify the qualities that the product must possess. These are things such as security, compatibility, performance requirements etc.

## Functional Vs Non-Functional Requirements

- Picture
- Sound
- Color
- AV In Out
- One remote
- 100 channels
- On off timer
- Parental lock
- Stereo sound
- User manual
- Broadcasting standards
- Color coding standards

Functional

- Sharper picture
- PIP
- Neutral Box Color
- Intelligent Channel search
- EPG
- Program Alert
- Ambi light
- TV Recording

Non- Functional



Copyright © Capgemini 2015. All Rights Reserved 9

## Types of Requirement

- **Functional Requirements**

- “Defines a function of a software system or its component. A function is described as a set of inputs, the behavior, and outputs.”

- **User Requirements - perceptions**

- Existing system functioning
  - Expected functioning
    - Eg: View Account Status, Pay bills online
  - Business rules – enablers
  - Business processes
    - Eg: Loan process- BGC, Document verification
  - Basic System operations and work flow
    - Eg: Online bill payment- View bill, pay bill, process payment

- **User Interfaces**

- What the user sees
    - Eg: Users-Screens /Reports



Copyright © Capgemini 2015. All Rights Reserved 10

A functional requirement defines a function of a software-system or its component. A function is described as a set of inputs, the behaviour, and outputs (see also software). Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that show how a use case is to be fulfilled.

## Types of Requirement

- Interface (or External API) Requirements
  - Requirements of interfaces to external systems
    - Eg: Online banking system- I/f with bank's database, payment interface
  - Specified using interface and protocol specifications
    - Eg: Bill payment through an SSL secured interface
  - Inputs or outputs
    - Eg: Bill payment system- Inputs from service providers database Output to customer's bank account and to service provider



Copyright © Capgemini 2015. All Rights Reserved 11

The aggregate of means by which the users/ systems interact with the users/ systems- a particular machine, device, computer program or other complex tools.

The user interface provides means of:

Input, allowing the users to manipulate a system

Output, allowing the system to produce the effects of the users' manipulation.

## Types of Requirements

- Constraints

- Architectural, design, or implementation related restrictions, limitations, controls, checks or decisions – to be treated as requirements
  - Eg: The information on the web interface should not be downloadable
- Physical constraints, business rules, data and content constraints, hardware constraints, software constraints, industry standards, legal and regulatory constraints and production environment constraints
  - Eg: The ATM system is dependent on the network in the village areas. Due to flood and consequent network jams there will be a delay in the transactions
  - Eg. The system should be functional on both the old and new servers and space needs to be maintained on both



Copyright © Capgemini 2015. All Rights Reserved 12

Project constraints have the potential to severely impact the ability to successfully provide a solution. Sometimes a constraint may look like an iceberg. On the surface, there may not appear to be much to it, but on close examination the iceberg may have a large potential impact. Maybe it will not be enough to sink the ship, but it may be enough to knock you way off course!

## Types of Requirements

- Data (Informational) Requirements
  - Requirements that specify some mandatory property of a data type or value.
  - Attributes of the objects- Data to create the object
    - Eg: Account creation- inputs required
  - Specified using logical data models, object models, or data dictionaries
    - Eg: User names should be alphanumeric



Copyright © Capgemini 2015. All Rights Reserved 13

## Do not overlook the “Non-Functional Requirements”!

- Non-Functional Requirements

- “Non-functional requirements specify the system’s‘ quality characteristics’ or ‘quality attributes’
- Often soft and not clearly defined - hard to measure and test against
- Examples: Security, Usability, Maintainability, Robustness and Performance”

- Examples:

- It should be easy to see the history of transactions
- The system should be available 99.9% of the time
- The system should use SSH public-key cryptography to authenticate the remote computer and allow the remote computer to authenticate the user, if necessary
- The system should be able to serve at the most 100 concurrent users



Copyright © Capgemini 2015. All Rights Reserved 14

## Non Functional Requirements: FURPS +

- Functionality
- Usability
- Reliability
- Performance
- Supportability
- + other such quality attributes



Copyright © Capgemini 2015. All Rights Reserved 15

### Non Functional Requirements: FURPS +

Functionality - The degree to which the software satisfies stated needs as indicated by the following sub attributes: suitability, accuracy, interoperability, compliance, and security.

Usability - The degree to which the software is easy to use as indicated by the following aspects: understandability, learnability, operability.

Reliability - The amount of time that the software is available for use as indicated by the following aspects: maturity, fault tolerance, recoverability.

Performance - is concerned with characteristics such as throughput, response time, recovery time, start-up time, and shutdown time.

Supportability - is concerned with characteristics such as testability, adaptability, maintainability, compatibility, configurability, installability, scalability, and localizability.

The "+" in FURPS+ also helps us to remember concerns such as:

Design requirements - Design constraints for designing system, for example, if you specify that a relational database is required, that's a design constraint.

Implementation requirements - Specifies or constrains the coding or construction of a system. Examples might include required standards, implementation languages, and resource limits.

Interface requirements - Specifies an external item with which a system must interact, or constraints on formats or other factors used within such an interaction.

Physical requirements - Specifies a physical constraint imposed on the system , shape, size, or weight.

## Non Functional Requirement - Usability

- Usability is the ease with which a user can learn to operate, prepare inputs for, and interpret outputs of system or component
- Usability requirements include:
  - Well-structured user manuals
  - Informative error messages
  - Help facilities
  - Well-formed graphical user interfaces
- Examples :
  - Aesthetics, Screen Navigation,
  - Readability of information,
  - Output format,
  - Labeling of fields,
  - Meaningful error messages,
  - Help files & User manual



Copyright © Capgemini 2015. All Rights Reserved 16

## Non Functional Requirement - Reliability

- Degree to which the system must behave in a user-acceptable fashion
- Constraints on the run-time behaviour of the system
- It includes the following:
  - Availability: Is the system available for service when requested by end users
  - Failure rate: How often does the system fail to deliver the service as expected by end-users
  - Accuracy: What precision is required in systems that produce numerical outputs
  - Eg: System uptime should be 99.9%, Exception handling (no hanging!!)



Copyright © Capgemini 2015. All Rights Reserved 17

## Non Functional Requirement - Performance

- Performance requirements concern the speed of operation of a system. Includes categories such as:
  - Response requirements (how quickly the system reacts to a user input)
  - Throughput requirements (how much the system can accomplish within a specified amount of time)
  - Capacity requirements (the number of customers or transactions that the system can accommodate)
  - Degradation mode (what is the acceptable mode of operation when the system has been degraded)
  - Eg. The system should be able to serve at the most 100 concurrent users , Response time for a transaction should be 2ms



Copyright © Capgemini 2015. All Rights Reserved 18

## Non Functional Requirement – Supportability

- Supportability requirements are concerned with the ease of modifications to the system to accommodate enhancements and repairs. Includes:
  - Adaptability: The ability to change the system to deal with additional application domain concepts
  - Maintainability: The ability to change the system to deal with new technology or to fix defects
  - Internationalization: The ability to change the system to deal with additional international conventions such as languages, or number formats, styles
  - Portability: The ease with which a system or component can be transferred from one environment to another
  - Eg. Application should be available for German & Japanese users, Personalization to include user details, configuring options, Application should work on Mac too



Copyright © Capgemini 2015. All Rights Reserved 19

## Other Non Functional Requirements: “+”

- Security requirements
- Safety requirements
- Scalability Requirements
- Configurability Requirements
- Data Retention requirements
- Disaster Recovery requirements
- Legal & Regulatory Requirements
- Licensing requirements



Copyright © Capgemini 2015. All Rights Reserved 20

## What is a “Good” Software Requirements?

- Specific
  - Correct: A true statement of what the requirement should do
  - Complete: Encompass all requirements of concern to the User
  - Unambiguous: Has only one interpretation
- Consistent: Does not conflict with other Requirements
- Verifiable: Can be tested to meet the Requirements
- Attainable: It should be within the scope of the project
- Understandable: Comprehendible by User, Business and Developers
- Detailed/ Granular: Granular to be implemented in test cases and design
- Explicit: Encompass all derived requirements
- Traceable: It should be possible to trace a component requirement to its source
- Manageable & Organized: Scalability and change management, should be structured



Copyright © Capgemini 2015. All Rights Reserved 21

## A “Good” Software Requirement – Correct & Complete

- Correct

- “A set of requirements is correct if and only if every requirement stated therein represents something required of the system to be built.” Davis (1993)
  - Verified by a subject-matter expert during a review

- Complete

- No requirements or necessary information should be missing
  - Completeness is also a desired characteristic of an individual requirement
  - Include all significant requirements, whether related to functionality, performance, design constraints, attributes, or external interfaces
  - A complete requirement leaves no room for guessing

- Example

- Correct Example - “On the event of power failure the battery backup must support normal system operations”
  - Incorrect Example - “On the event of power failure the battery backup must support normal system operations for 30 minutes”



Copyright © Capgemini 2015. All Rights Reserved 22

### Characteristics of “GOOD” Requirements

**Correct** : Does every requirement state something required of the system? “A set of requirements is correct if and only if every requirement stated therein represents something required of the system to be built.” Davis (1993) It is not possible to determine if a requirement is correct simply by reading the requirement. The correctness is verified by a subject-matter expert during a review.

**Complete** : Does the set of requirements include all significant requirements, whether related to functionality, performance, design constraints, attributes, or external interfaces? Have the expected ranges of input values in all possible scenarios been identified and addressed? Have responses been included to both valid and invalid input values? Do all figures, tables, and diagrams include full labels, references, and definitions of all terms and units of measure?

## A “Good” Software Requirement – Unambiguous

- **Unambiguous**

- The reader of a requirement statement should be able to draw only one interpretation of it
- At the same time multiple readers of a requirement should arrive at the same interpretation
- The natural languages like English is highly prone to ambiguity
- Write each requirement in concise, simple, straightforward language of the user domain
- Incorrect and ambiguous requirements terms cause budget and schedule problems
- Poor writing can obscure the most necessary requirement
- Example
  - **Incorrect Example** – “The system should not accept password longer than 8 characters”
    - The above stated requirement can have multiple interpretation as given below
    - The system should not allow the user to enter more than 8 characters in the password field
    - The system should truncate the entered data to 8 characters
    - The system should display an error message if the user enters more than 8 characters in the password field
  - **Correct Example** – “The system should not accept passwords longer than 8 characters in the password field. If the user enters more than 8 characters while entering the password, an error message should prompt the user to correct the same.”



Copyright © Capgemini 2015. All Rights Reserved 23

**Unambiguous:** Each requirement statement should have only one interpretation, and each requirement should be specified in a coherent, easy-to-understand manner.

## A “Good” Software Requirement – Consistent

- **Consistent**

- A consistent requirement does not conflict with other requirements in the requirement specification
- Vision document, the use-case model, and the Supplementary Specifications should be in sync & not conflicting
- The requirements also do not conflict with higher-level requirements including business, user, or system level requirements
- Terminology should also be used consistently within the document:
  - A word has the same meaning every time it is used
  - Two different words are not used to mean the same thing
- Example
  - REQ1 – “The birth date should be entered in mm/dd/yyyy format”
  - REQ2 - “The birth date should be entered in dd/mm/yyyy format”
- Sometimes it is possible to resolve the conflict by analyzing the conditions under which the requirement takes place
- The above requirements can be rewritten as given below:
  - REQ1 – “For the US users the birth date should be entered in mm/dd/yyyy format”
  - REQ2 – “For the French users the birth date should be entered in mm/dd/yyyy format”
  - The user can withdraw a given amount of money from the account, but not more than Rs 15,000 a day
  - The user can withdraw a given amount of money from the account, not more than three withdrawals a day



Copyright © Capgemini 2015. All Rights Reserved 24

**Consistent :** Is it internally consistent, with no subset of individual requirements described which are in conflict? (Examples: Vision document, the use-case model, and the Supplementary Specifications.)

## A “Good” Software Requirement – Verifiable

- **Verifiable**

- A verifiable requirement is stated in such a way that it can be tested by inspection, analysis or demonstration
- The testers should be able to verify whether the requirement is implemented correctly
- To be verifiable, the requirement should be clear, precise and unambiguous
- A verifiable requirement makes it possible to evaluate whether the system meet the requirements
- **Example**
  - Incorrect Example – “The applications should be user friendly”
  - Correct Example – “The UI of the application should be menu driven. It should provide dialog boxes, radio buttons, dropdown list boxes and numeric updown controls for user input”



Copyright © Capgemini 2015. All Rights Reserved 25

**Verifiable** : Is every requirement stated verifiable? Is there some finite cost-effective process with which a person or machine can check that the software product meets the requirement?

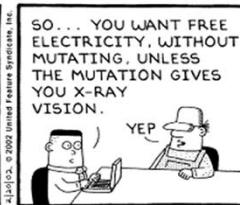
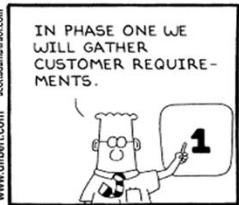
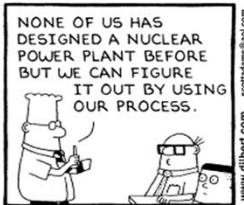
## A “Good” Software Requirement – Attainable

- Attainable

- The requirement can be implemented using available technologies, techniques, tools, resources, and personnel within the specified cost and schedule constraints

- Example

- Incorrect Example – “The replacement control system should be installed with no interruption to production”
- Correct Example – “The replacement control system should be installed causing not more than 2 days of production interruption”



Copyright © 2002 United Feature Syndicate, Inc.

Copyright © Capgemini 2015. All Rights Reserved. 26

**Attainable :** The requirement should be doable within existing constraints such as time, money, and available resources. An attainable requirement can be:

- Met using existing technology
- Achieved within the budget
- Met within the schedule
- Is something the organization has the necessary skills to utilize

## A “Good” Software Requirement – Understandable, Granular, Explicit

- Comprehended by users and developers – Same meaning in the same context
- Granularity should be such that it can be implemented in design and test cases
- All requirements should be explicit: NO ASSUMPTIONS should be made!



Copyright © Capgemini 2015. All Rights Reserved 27

**Understandable:** Comprehended by users and developers the same way, with the same intent. Too often, requirements (especially derived requirements) are specified by developers who use their technical jargon that is not understandable to other stakeholders, especially customers, users, and managers. But individual requirements should be oriented around the needs of the customers and users if they are to be understandable and validable.

## A “Good” Software Requirement – Traceable

- Each requirement should be expressed only once and should not overlap with another requirement
- A traceable requirement has a unique identity or number
- It can be easily traced through to specification, design and testing
- Each requirement should be traceable back to its source
- It should also be specified in a manner that allows traceability forward into the design, implementation, and testing



Copyright © Capgemini 2015. All Rights Reserved 28

### Traceable :

- a. Does each requirement have a clear identifier?
- b. Is it distinguishable from non-essential statements in the requirements set?
- c. Is the origin of each requirement clear?
- d. Is backward traceability maintained by explicitly referencing earlier artifacts?
- e. Is a reasonable amount of forward traceability maintained to artifacts spawned by the requirements set? For example, test cases.

## A “Good” Software Requirement – Manageable & Organized

- Keep sentences short, numbered and specific
- Avoid Paragraphs
- Define terms in a glossary
- Document in a hierarchical, structured form
- Avoid redundancies

SR 1 – The User can access the most recent invoice through the email notification or the website

SR 1.1 – The User will be provided a link on the home page to view ‘Most Recent Invoice’

SRS 1.1.1 - On clicking the link will be taken directly to the invoice generation page

SR 1.2 – The User can also access the link from the email notification received by the User to the preferred email ID on or after the Invoice Generation date

SR1.2.1 - While clicking the link via email the User will have to login to the system, and on successful login will be taken directly to the invoice generation page



Copyright © Capgemini 2015. All Rights Reserved 29

## Summary

- In this lesson, you have learnt:
  - Evolution of Requirements
  - Who provides requirements?
  - Functional Requirements
  - Non-Functional Requirements
  - Comparison between functional and non-functional requirements
  - Characteristics of good software requirement



## Review Question

- Question 1: Evolution of Requirements involves the translation of stakeholder requests into a set of system features.
  - True/ False
- Question 2: Reliability is a degree to which the system must behave in a user-acceptable fashion.
  - True/ False
- Question 3: \_\_\_\_\_ define how software behaves to meet user needs.
- Question 4: Which of the following characteristics justifies the statement that "The reader of a requirement statement should be able to draw only one interpretation of it"?
  - Option 1: Unambiguous
  - Option 2: Correct
  - Option 3: Consistent
  - Option 4: Verifiable



Copyright © Capgemini 2015. All Rights Reserved 31

## Review Question

- Question 5: Which of the following non functional requirements can ensure that the application should be available for German & Japanese users?
  - Option 1: Internationalization
  - Option 2: Security
  - Option 3: Portability
  - Option 4: Adaptability



# **Requirements Validation and functional decomposition**

Lesson 3: Requirements  
Itemization

## Lesson Objectives

- To understand the following topics:
  - Requirements Itemization - How?
  - Application Background
  - Assumptions
  - Explicit requirements
  - Implicit requirements
  - Interface Requirements
  - Requirements Analysis to Test Scenarios
  - Testable Items
  - Non Testable Items
  - Error Conditions
  - Application Invocation
  - Application Termination



## Lesson Objectives

- To understand the following topics:
  - File Handling
  - Requirement Prioritization
  - Case Study – Equipment Tracking System
  - Template – 1 Equipment tracking System
  - Template – 2 Equipment tracking System
  - Summary
  - Review Questions



## Requirements Itemization - How?

- Simplifies the requirements for better understanding
- Helps in identifying different Test Scenarios
- Helps in identifying testable items for the application
- The SRS document might contain non functional (e.g. performance) requirements along with Functional requirements
- Non functional requirements itemization must also be carried out



Copyright © Capgemini 2015. All Rights Reserved 4

Requirement Itemization is required for following :

To simplify the requirements for better understanding of the system user test  
To identify different Test scenarios  
To identify testable items for the application

Requirement Itemization needs to be done for both functional and non-functional requirements.

## Requirements Itemization - How?

- Application Background
  - Identify the background of the Application Under Test
- Assumptions
  - Identify the assumptions for the Application Under Test
- Requirements Analysis and Prioritization
  - Analyze and prioritize the requirements
- Creating Test Scenarios from Requirements
  - Create Test scenarios from the requirements given



Copyright © Capgemini 2015. All Rights Reserved 5

When splitting the requirements in smaller components, they can be split on the following categories:

Background: Background of the application under test

Assumptions: Assumption those can be used while testing the application

Requirements: These can be divided further into Functional Requirements, Non Functional requirements and design objectives

Creating Test Scenarios from Requirements

Error conditions

Application Invocation

Application Termination

File Handling

## Requirements Itemization - How?

- Error Conditions
  - Identify expected and unexpected error conditions
- Application Invocation
  - Identify the different ways to start the application
- Application Termination
  - Identify the different ways to end the application
- File Handling
  - Analyze how file handling takes place in the application



Copyright © Capgemini 2015. All Rights Reserved 6

Error conditions : Exceptions to be considered in the application

Application Invocation : Different ways to invoke the application

Application Termination : Different ways of terminating the application

File Handling : Different files that getting referred, used and updated through application

## Application Background

- System Requirements for the application
  - Special requirements regarding operating system, size of the memory required for application to run etc.
- Module Under Test
  - Modules to be tested and their scope in the application
- System Flow
  - How the data will flow in the application and how different modules in the application are integrated
- Installation requirements
  - Details of the test environment for Installation of the software
- Security Information – User Access rights
  - This will include User access rights, authentications etc. What are the types of Users for the system under test and what will be rights given to the users



Copyright © Capgemini 2015. All Rights Reserved 7

System Requirements for the application :

For any application there could be special system requirements. These can be regarding operating system, size of the memory required to run the application etc.

Module under test :

Find out which are the modules to be tested and their scope the application.

Flow of the system :

How the data will flow in the system and how the modules are interconnected to each other?

Installation Requirements :

The details on how the application will be installed for testing. Details of the test environment for Installation of the software.

Security information :

This will include User access rights, authentications etc. What are the types of Users for the system under test and what will be rights given to the users.

## Assumptions

- Assumptions may be detailed in the requirements document
- These assumptions could be related to the users, functional specifications, input data etc.
- E.g. - Assumption in an equipment tracking system
  - Equipment entry is already in the system as Purchase phase is over
  - Equipment 'Use status' can be 'In Use' if it is installed and assigned to some user/Department or it can be 'In stock' if it is not put in Use



Copyright © Capgemini 2015. All Rights Reserved 8

Assumptions are mentioned in the SRS

These are the certain conditions or status, if it is present or existing, then the application under the test can perform its functionality.

In other words, it can be termed as pre-requisites for the application to carry out its functionality.

## Explicit requirements

- Explicit Requirements are requirements
  - Which are explicitly stated in the SRS
  - These are requirements that are received from the customer
    - E.g. ATM machine shall validate PIN number entered by the customer



Copyright © Capgemini 2015. All Rights Reserved 9

The Explicit requirements are the ones which the customer has specifically asked to deliver, it is the expected output or result of the system.

E.g. Every user should be validated before he accesses the Accounting system by the username/password protection. This is specified by the customer, hence it is a explicit requirement.

## Implicit requirements

- Implicit Requirements are also termed as derived requirements
- These requirements are generated based on the functional requirements received from the customer
- These are not explicitly given by the customer
- E.g. Money can be withdrawn only if there is enough Cash available in ATM machine



Copyright © Capgemini 2015. All Rights Reserved 10

This requirement is not specified in the SRS by the customer. But, it has to be validated for proper functioning of the system. So, this is a implicit requirement.

## Interface Requirements

- Interface requirements are requirements related to the Internal and External interfaces of the system
  - E.g. Displaying updating balance
- Internal interfaces are interfaces between internal modules of the System under test
  - E.g. Balance is updated after credit or debit of the account
- External Interfaces relate to the given system's communication with other systems.
  - E.g. Transaction record is printed if requested



Copyright © Capgemini 2015. All Rights Reserved 11

Examples of interfaces are:

Person-machine interface which includes operating system, language compilers and I-O facilities

Communication interfaces which include transmission of information between the computer and remote equipments

Program interfaces which include exchange of information whether on the same computer or distributed across multiple tiers application architecture

## Requirements Analysis to Test Scenarios

- When analyzing the requirements, different Test Scenarios are identified for the requirement
  - E.g. Valid User Login to Equipment Tracking System (Manage Equipment) & Update the Equipments



Copyright © Capgemini 2015. All Rights Reserved 12

All requirements are analyzed from Tester's point of view. First different test scenarios are identified from the requirements.

Each requirement is analyzed whether it is testable or not.

## Requirements Analysis to Test Scenarios

- Determine whether the requirement is Testable or not
- Queries should be routed to either the Client or to the Subject Matter expert(s) or domain specialist(s)



Copyright © Capgemini 2015. All Rights Reserved 13

All functional requirements are testable requirements. Depending on the scope of the project, we will need to decide whether the non-functional requirements can be tested or not.

There are bound to have some queries while analyzing the requirements document. These queries need to be routed to the concerned person. The queries about functionality, technical queries should be sent to the client or SME (subject matter expert). SME is technical expert on the project and he has domain knowledge. The queries about functionality should be sent to the client, technical queries should be sent to SME. A separate sheet is maintained for all types of queries and those are tracked.

## Testable Items

- In the requirement document there are most of the requirements that can be tested
- These are the requirements that are not frequently changing
- These can be data validation, form validation, field validation, & so on
- For E.g. integer value validation can be tested easily



Copyright © Capgemini 2015. All Rights Reserved 14

## Non Testable Items

- In the requirements documents there can be some requirements which can not be tested
- These requirements can be performance related and hence they may not be in the scope of functional testing
- There can be some requirements which changes dynamically on web applications
- Such requirements may be non testable requirements
- For example if the session variable for the URL in web application is stored in some HTTP file and it is changing dynamically
- Some implicit requirements where query needs to be raised may be non testable in the initial stage, but they may be testable items when the query is resolved
  - For example: If it is not stated who are the users for the application and what are the access rights for those users in the requirements document then such items could not be tested unless the query is resolved. Once the query is resolved these non testable items will be converted into testable items.



Copyright © Capgemini 2015. All Rights Reserved 15

## Error Conditions

- Expected Error Conditions - Error conditions are stated in the requirements document. These are expected errors for any Erroneous input. E.g. Invalid user name or password is entered in the login screen
- Unexpected Error Conditions - These are unexpected error Conditions like Server time out or link down. While analyzing requirements we should document these error conditions separately



Copyright © Capgemini 2015. All Rights Reserved 16

## Application Invocation

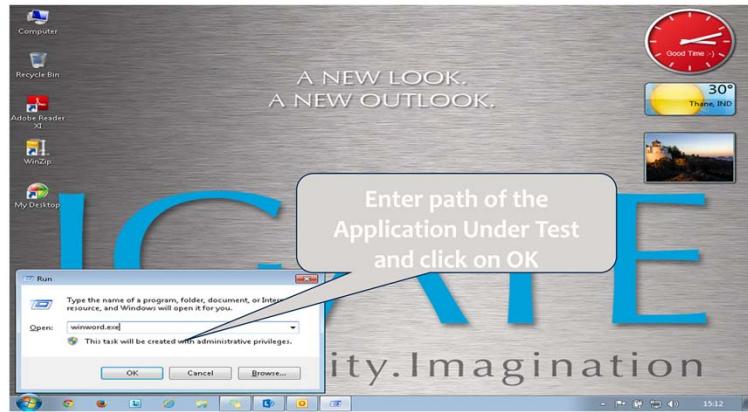
- There can be number of ways in which an application can be invoked. E.g. through Windows Explorer, through Command Line, through a web form etc.
- Start up process involves the files required to start the application or the process involved in the application startup



Copyright © Capgemini 2015. All Rights Reserved 17

## Application Invocation

- Let us see how to invoke the AUT using Run option of Windows Explorer



## Application Termination

- There can be number of ways in which the application can be terminated. E.g. Using the mouse, using hot keys etc.
- Termination of application will involve shutdown process for the application



Copyright © Capgemini 2015. All Rights Reserved 19

## File Handling

- Different types of files are created / updated within the application
  - Some files may be required before the application is started
  - Some files may be required to be present while the application is being invoked
  - Some files may be created or updated as the application executes
  - Some files may need to be uploaded or downloaded for application execution



Copyright © Capgemini 2015. All Rights Reserved 20

There are some files which are required to be present while application Invocation. If any of the files is not present or is invalid then Application will not invoke.

Similarly there are some files which are created or updated as the application executes. All these files must be separately documented in File Handling part from the requirements document.

## Introduction to Requirement Prioritization

- Most of the software development projects consists of many candidate requirements that needs to be realized within the project constraints like time, cost and other resources
- Requirement Prioritization helps to identify the most essential requirements from this set by distinguishing the critical few from the trivial many
- When customer expectations are high, timelines are short and resources are limited, we need to make sure that the software system features all required functionalities
- Requirement prioritization can help project teams to attain above mentioned goal by making a right choice of requirements at right time
- Requirement Prioritization is the process of managing the relative importance and urgency of different requirements to cope with limited resources of projects
- Prioritization must be done in collaboration with stakeholders



Copyright © Capgemini 2015. All Rights Reserved 21

### Introduction to Requirement Prioritization

Most of the software development projects consists of many candidate requirements that needs to be realized within the time and cost constraint. However, if the development team cannot deliver every requirement by the scheduled initial delivery date, the project stakeholders must agree on which subset to implement first. Requirement prioritization or establishing relative priorities of the requested functionalities is an essential activity that needs to be carried out by every project, especially those are with limited resources and cost.

Requirement prioritization must be done in collaboration with stakeholders – Customers, Product Owner and Users. It is recommended to have developers and stakeholders working together in prioritization process as they always have conflicting views and needs about requirement and their importance. A common trap is to let the stakeholders choose the priorities without any guidance. In those situations, the stakeholders likely tag most requirements as being critical. To avoid this situation the business analyst must guide the customers through the proper requirement prioritization process.

The typical participants in the prioritization process include:  
The project manager  
Key customer representatives  
Development representatives, such as team technical leads

## Why Requirement Prioritization?

- Following are some of the aspects that signifies the importance of prioritization process in projects
  - Varied levels of importance : Not all requirements are equally important, and the many different stakeholders in the system typically will not agree as to which requirements are most important.
  - Limited Project Resources : All projects have limited resources in terms of budget, staff, and schedule. It is impractical to implement all the requirements during the system's current release. The incremental development approach is preferred to ensure that even trivial requirements are realized during product development.
  - Long Running Projects : Incrementally developed systems requires months to years to develop during which requirements are subject to significant iteration as the business environment changes, business needs change, and new requirements are identified. These situations further increases the need for prioritization.
  - Penalty : It is possible to evaluate the penalty that is introduced if some of the requirements are not fulfilled. For example, non-conform to a standard could incur a high penalty even if it is of low importance for the customer.



Copyright © Capgemini 2015. All Rights Reserved 22

### Why Requirement Prioritization?

The quality of the software system is often determined by its capability of satisfying the needs of the customers & users. Therefore, eliciting and specifying the correct set of requirements and planning appropriate suitable releases with the right functionality is a major step towards the success of a project or product. Prioritization helps to identify the most valuable requirements and implement them first. One of the important feature of a good requirement is having to be explicitly prioritized. When customer expectations are relatively high and project team required to ensure that within limited timelines and resources the software system exhibits the most essential functionalities. Establishing priorities for important functionalities lets you order the development activities to provide the greatest product value at the lowest cost.

The slide above throws light on some of the project facts that make the prioritization of requirements a critically important part of requirements analysis that every requirements engineer must perform.

## Benefits of Requirement Prioritization

- Properly prioritizing requirements provides the following significant benefits to the project:
  - It enables the development team and stakeholders to plan and design optimal set of software requirements for implementation in successive releases
  - It helps project team to trade off desired project scope against sometimes conflicting constraints such as schedule, budget, resources, time to market, and quality
  - Helps in choosing only a subset of the requirements and still produce a system that will satisfy the customer
  - Ensures increase in customer satisfaction by implementing & delivering customer's most important requirements first
  - It lessens the risk of project cancellation as valuable progress is being demonstrated with each increment
  - Prioritizing requirements is a good way to force stakeholders to address all requirements and not just their own
  - It also helps in minimizing rework and schedule slippage



Copyright © Capgemini 2015. All Rights Reserved 23

## Requirement Priority Scales

- The effective requirement prioritization demands the use of a ranking scheme
- The number of different scales are used in practice to indicate the relative importance of a requirement like categorical scales, linear and non-linear numeric scales
- A project team is responsible for deciding the ranking scheme
- Initially, a simple categorical scale can be used to triage requirements
- Then a numeric scale can be applied to further prioritize the requirements
- Once the requirements are prioritized the list is ordered and implementation starts with the most important ones



Copyright © Capgemini 2015. All Rights Reserved 24

## Requirement Priority Scales



- High, Medium, Low
- Critical, Important , Desirable
- Essential, Conditional, Optional

- Range of numeric values
- 1-10 or 1-100

- Modified Fibonacci – 1,2,3,5,8,13,20,30

## Requirement Priority Scales

- It is important for all stakeholders to understand the meaning of each priority value
- For a numeric scale, a small value means a low priority reduced necessity and less urgency , while a large value indicates a high priority necessary and urgent
- For categorical scales, a definition of each categorical value needs to be established

Priority	Meaning
High/Critical/Essential	A critical requirement without which the product is not acceptable to the stakeholders
Medium/Important/Desirable	A necessary but deferrable requirement which makes the product less usable but still functional
Low/Conditional/Optional	A nice feature to have if there are resources but the product functions well without it



Copyright © Capgemini 2015. All Rights Reserved 26

## Requirement Prioritization Process Guidelines

- The requirement team needs to convince the stakeholders regarding the importance of prioritizing requirements
- The stakeholders should be trained on requirement prioritization process
- Categorize raw potential requirements during requirement elicitation phase into actual requirements and desirable requirements so that the actual requirements can be prioritized
- During requirement analysis, requirement team should work closely with stakeholders to prioritize the actual requirements
- This includes negotiation with the stakeholders to develop a consensus and validation of the resultant priorities with them
- The development team that must actually implement the requirements creates and records realistic estimates of the effort required to implement each requirement
- The requirement team with management & development team schedules development based on priorities of the requirements



Copyright © Capgemini 2015. All Rights Reserved 27

## Requirement Prioritization Process Guidelines

- During requirements management, the requirements team should work with the requirements stakeholders to maintain the requirements parameters as they change
- This will typically include storing the priority as metadata in the requirements repository, and then updating the value of the priority as it changes



Copyright © Capgemini 2015. All Rights Reserved 28

## Prioritization Techniques

- Following are some of the known requirement prioritization techniques used to determine, negotiate, and develop a consensus regarding the priorities of the requirements:
  - Business Case Analysis / Return On Investment (ROI) estimation
  - Pair-wise comparisons
  - Prioritization working groups
  - Scale of 1-to-10 rankings
  - Voting schemes (e.g., give each stakeholder a specific number of votes to distribute amongst the requirements or classes of requirements being prioritized)
  - Weightings (e.g., weight the votes of different stakeholders)
  - Value-Based Software Engineering [Boehm 2003]
  - WIN-WIN [Boehm 2001]
  - Quality Function Deployment (QFD)



Copyright © Capgemini 2015. All Rights Reserved 29

## Case Study – Equipment Tracking System

- Case Study for requirements Itemization for an equipment tracking system



Copyright © Capgemini 2015. All Rights Reserved 30

## Case Study - Step 1

- Separate the application background from Requirements
- Document. Also separate module under test and flow of the System
  - E.g. Background Of the application
  - System is Equipment tracking
  - The functionality is available on fat Client and on web
- Module Under test - Manage equipments
- Flow of the system - Purchase equipment -> Manage Equipment -> Track financials -> Retire equipment



Copyright © Capgemini 2015. All Rights Reserved 31

## Case Study - Step 2

- List down assumptions from the requirements document
- specified for the system under test
- E.g. Assumptions:
  - Equipment entry is already in the system as Purchase phase is over
  - Equipment 'Use status' can be 'In Use' if it is installed and assigned to some user/Department or it can be 'In stock' if it is not put in Use



Copyright © Capgemini 2015. All Rights Reserved 32

## Case Study - Step 3

- Identify the following:
  - Different Scenarios (Update, Print, Report etc.)
  - Testability of the scenarios and
  - Type of the requirement.
    - (E.g. on the next slide)



Copyright © Capgemini 2015. All Rights Reserved 33

## Case Study - Step 3

Req. Id.	Scenarios	Description	Post Condition	Type	Testable Y /N
1	Login	Only authorized users and system can update equipment record	Functionality is accessible	Explicit	Yes
2	Login	Unauthorized users should not be able to login to the system	Functionality is not accessible	Implicit	Yes
3	Verify	Check for the equipment type properties for updating	Properties are checked	Explicit	Yes
4	Update	Updates when end date is not expired. Only valid locations and valid users are assigned to equipment record.	Updating is done	Explicit	Yes
5	Update	Update equipment record when end date is expired.	Updating is done	Explicit	Yes



Copyright © Capgemini 2015. All Rights Reserved 34

## Case Study - Step 4

- While identifying scenarios you need to maintain a query log
- Separate queries for Client and for subject matter experts

Client	Subject matter Expert
Which functionality will be on FAT client and which on web?	How the users are categorized into : Inventory personnel, equipment auditors, service personnel, maintenance personal and Equipment Tracking personnel
What is an external system?	There is contradiction between statements 3.2.1 and 3.2.2. Equipment tag is numeric or character?
	If the equipment tag is all numeric then how 'AD' will be present in position 1 and 2 which needs to be removed



## Case Study Step 5

- Document the Expected and Unexpected Error conditions specified in requirements document

Req. Id.	Scenarios	Description	Post Condition	Type	Testable Y / N
9.4	Check errors generated	Check error generated when equipment tag is not found in equipment tracking system	Error generated is verified	Explicit	Yes
		Check error generated when last scan date stored in equipment tracking system is greater than last scan date stored in comp track system	Error generated is verified	Explicit	Yes



Copyright © Capgemini 2015. All Rights Reserved 36

## Template - 1

- This template consists of only one sheet
- Requirements are separated as per different categories. E.g.
  - Invocation, Termination, File Handling, error handling, system requirements, Security requirements, explicit, Implicit, interface requirements etc.
- Each requirement Id will have requirements converted into test conditions and its post conditions
- Query log will be maintained separately



Copyright © Capgemini 2015. All Rights Reserved 37

## Template - 1

Sr. No	Req. Id	Scenario	Requirement	Expected Result	Category	Testable Y / N
1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						
17						
18						
19						
20						
21						
22						
23						
24						
25						
26						
27						
28						
29						
30						
31						
32						
33						
34						
35						

Capgemini  
CONSULTING TECHNOLOGY OUTSOURCING

Copyright © Capgemini 2015. All Rights Reserved 38

## Template – 1 Equipment tracking System

Sr.No	Req. ID	Scenarios	Requirement	Expected Result	Category	Testable Y/N
1	2	Update	Only selected user roles (list given) can update equipment record	Selected user are able to update equipment record	Explicit	Y
2		Update	The user roles which are not in the list, can not update equipment records	Users those are not part of the given list of authorized users are not able to update equipment record	Implicit	Y
3	3.5	Query	Users can query for an equipment by "User ID"	Equipment can be queried based on "User ID"	Explicit	Y
4	3.6	Query	Users can query for an equipment by "Location" selected from a list	Equipment can be queried based on "Location" selected from a list	Explicit	Y
5	4, 4.1 to 4.6	Query	A list of equipment matching the query criteria (result set) should be displayed containing following fields : Equipment Tag, Quantity, Seq Number, User ID, Location, Equipment Type	The list of equipments displayed with mentioned fields	Explicit	Y
6	5.1	Update	Users can modify equipment record of their state only	Users are able modify equipment record successfully of their state only	Explicit	Y



## Template – 1 Equipment tracking System

Sr.No	Req. ID	Scenarios	Requirement	Expected Result	Category	Testable Y/N
7		Update	Users can not modify equipment record of other state	Users are not able modify equipment record of other state	Implicit	Y
8	5.2	Update	For retired equipments : only comments field can be modified	Updation can be done to only comments field for a retired equipment	Explicit	Y
9		Update	For retired equipments all fields can not be modified except comments	Updation of all fields except comments field is restricted for a retired equipment	Implicit	Y
10	10.1 to 10.4	Report	CSV file (With given fields) should be created for equipments received in current month, excluding any "Retired" equipments	CSV file created successfully for equipments received in current month, excluding any "Retired" equipments	Explicit	Y
11		Report	CSV file (With given fields) should not contain data for equipments received in non-current month and including any "Retired" equipments	CSV file is not getting created for equipments received in non current month, including any "Retired" equipments	Implicit	Y



Copyright © Capgemini 2015. All Rights Reserved 40

## Query Log – Equipment tracking System

Client	SME
What is authorized system? How those will update equipment record?	What are most recent rules about Seq Number? What is the format, Number or Character?
	What is the meaning of - Manually update specified data for a single equipment record?
	What is the meaning of - Manually update specified data for multiple equipments record simultaneously?
	What is unassociated equipments identified as "Spare Part" ?
	How Seq Number is related to the equipment type?



Copyright © Capgemini 2015. All Rights Reserved 41

## Error Conditions – Equipment tracking System

Req. ID	Scenarios	Description	Category	Post Condition	Testable Y/N
8	Update	Check error generated when the external system fails while updating equipment record (Unexpected Error)	Implicit		N
8		Check error generated when the Comp Track system fails while receiving automatic updates (Unexpected Error)	Implicit		N
		Server time-out, Link down	Implicit		Y
7		Check error generated when various validations fail during updation (Unexpected Error)	Implicit		Y



Copyright © Capgemini 2015. All Rights Reserved 42

## Pros and Cons (Template - 1)

- Pros

- Useful when requirements are for smaller projects
- Requirements can be filtered category wise for creating test cases
- Only one sheet needs to be maintained

- CONS

- Can not be used for larger project requirements



Copyright © Capgemini 2015. All Rights Reserved 43

## Template - 2

- Separate sheets will be maintained for different categories
- For each category, the Requirement Id with the test conditions and post conditions is written
- Query sheet will be maintained separately



Copyright © Capgemini 2015. All Rights Reserved 44

## Template – 2

The screenshot shows a Microsoft Excel spreadsheet titled "Req Item Template [Compatibility Mode] - Microsoft Excel". The spreadsheet has a header row with columns for "Sr. No", "Req. Id", "Scenario", "Description", "Post Condition", and "Testable Y / N". Rows 3 through 36 are empty. The Capgemini logo is visible at the bottom left, and copyright information is at the bottom right.

## Template – 2 Equipment tracking System

- For a case study given in lab book, in Security tab, following will be the input

Sr.	Req. ID	Scenario	Description	Post Condition	Testable Y/N
1	1	Login	Authorized users and systems must have the ability to record updates to equipment records	Records are updated	Y



Copyright © Capgemini 2015. All Rights Reserved 46

## Template – 2 Equipment tracking System

- For Explicit tab, all the mentioned functionalities will be included

Sr.	Req. ID	Scenario	Description	Post Condition	Testable Y/N
1	5.1	Update	Users are restricted to modify equipment records only within their state	Records are updated	Y



Copyright © Capgemini 2015. All Rights Reserved 47

## Template – 2 Equipment tracking System

- For Implicit tab, all the derived functionalities will be included

Sr.	Req. ID	Scenario	Description	Post Condition	Testable Y/N
1	5.1	Update	Users are modifying equipment records other than their state	Records are not getting modified	Y



Copyright © Capgemini 2015. All Rights Reserved 48

## Template – 2 Equipment tracking System

- For Invocation tab, all the ways to invoke an application are to be captured

Sr.	Req. ID	Scenario	Description	Post Condition	Testable Y/N
1		Application Invocation	Through IE	Application getting invoked	Y



Copyright © Capgemini 2015. All Rights Reserved 49

## Template – 2 Equipment tracking System

- For Termination tab, all the ways to close an application are to be captured

Sr.	Req. ID	Scenario	Description	Post Condition	Testable Y/N
1		Application Termination	Click on Close button (X) of an application	Application getting closed	Y



Copyright © Capgemini 2015. All Rights Reserved 50

## Template – 2 Equipment tracking System

- File handling tab will be taking care of all the files getting used through out the application
- Error handling tab will be considering all the fatal and non fatal error condition for an application
- System tab will be talking about all the system specification requirements as per SRS
- Query tab will be taking care of queries to be forwarded to client as well as SME



Copyright © Capgemini 2015. All Rights Reserved 51

## Pros and Cons (Template - 2)

- Pros

- Useful when requirements are for larger projects
- Separate sheets are available for different categories

- Cons

- Multiple sheets need to be maintained
- Time consuming
- Some requirements might get repeated if they fall in two or more different categories



Copyright © Capgemini 2015. All Rights Reserved 52

## Case Study: Query Log

### ■ Equipment Tracking System

Client	SME
What is authorized system? How these will update equipment record?	What are most recent rules about Seq Number? What is the format, Num or Char?
	What is the meaning of manually update specified data for a single equipment record?
	What is the meaning of manually update specified data for a multiple equipment record?
	What is unassociated equipment identified as spare parts?
	How Seq Number is related to the equipment type?



Copyright © Capgemini 2015. All Rights Reserved 53

## Summary

- In this lesson, you have learnt:
  - How to perform requirement itemization?
  - Requirements Analysis to Test Scenarios
  - Testable Items
  - Non Testable Items
  - Requirement Prioritization
  - Templates used in requirement itemization



## Review Question

- Question 1: Interface requirements are requirements related to the Internal and External interfaces of the system.
  - True/ False
- Question 2: Requirement Prioritization is the process of managing the relative importance and urgency of different requirements to cope with limited resources of projects.
  - True/ False
- Question 3: \_\_\_\_\_ are also termed as derived requirements.
- Question 4: \_\_\_\_\_ are requirements related to the Internal and External interfaces of the system.



## Review Question

- Question 5: Which of the following are requirement prioritization techniques?
  - Option 1: Pair-wise comparisons
  - Option 2: Prioritization working groups
  - Option 3: Scale of 1-to-10 rankings
  - Option 4: All of the above



## **Requirements Validation and Functional Decomposition**

Lesson 4: Requirements  
Gathering – Challenges &  
Techniques

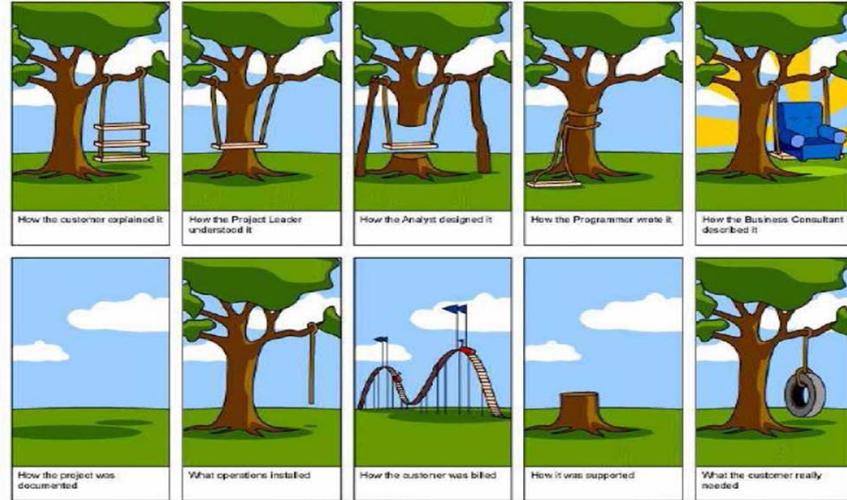
## Lesson Objectives

- To understand the following topics:
  - Requirements Gathering – A Typical Illustration
  - Requirement Gathering Patterns
  - Challenges in Requirements
  - Challenges – Clarity of requirements
  - Challenges – Communication
  - Ambiguity From a Requirements Perspective - Pitfalls of the English language
  - Ambiguity and Pitfalls
  - Ambiguity Checklist
  - Ambiguity Review
  - Requirement Gathering - Skills Required
  - Tips to Requirement Gathering
  - Identification and Verification of Requirements
  - Summary
  - Review Questions



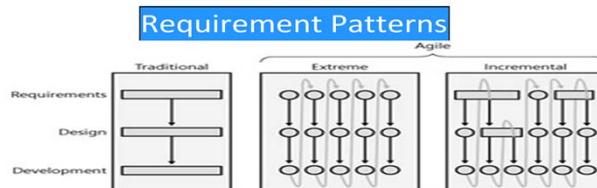
Copyright © Capgemini 2015. All Rights Reserved. 2.

## Requirements Gathering – A Typical Illustration



## Requirement Gathering Patterns

- Traditional approach
  - Requirements are specified in detail and passes thru multiple reviews and sign-offs
- Extreme approach
  - Strives to make product/application ASAP and generally requirements needs to be elaborated
- Incremental approach
  - Some of the requirements are detailed up-front and some as the project proceeds



Copyright © Capgemini 2015. All Rights Reserved. 4

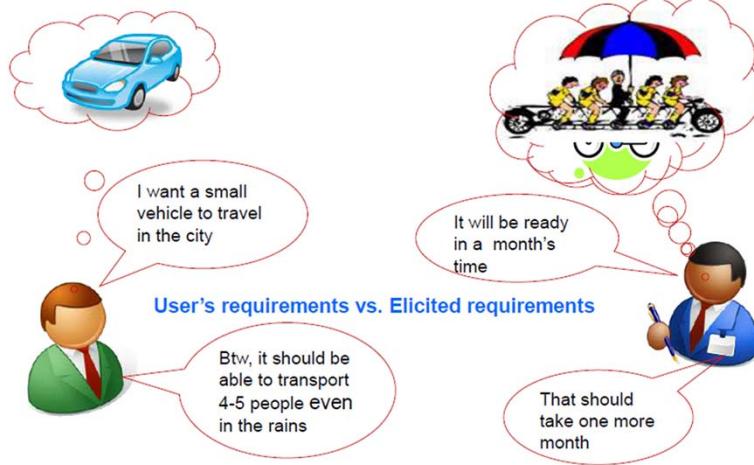
## Challenges in Requirements

- User Involvement
  - “It is month end, quarter end ,year end. I have to do all my reports so
  - can’t spend time on requirements”
- Customer Expectations
  - Unreasonable, Infeasible, Conflicting in many occasions
- Scope and Vision not clearly defined
  - All requirements are critical, no priority is defined
- Improper Change Management
  - New requirements get added in the middle of the project
  - Users/customers are busy and not available to specify requirements
  - Signed-off requirements keep changing
- Functionality built, rarely or never used



Copyright © Capgemini 2015. All Rights Reserved. 5

## Challenges – Clarity of requirements



## Challenges – Communication

- Language barriers
  - Have interpreter
    - E.g. Japanese projects
  - Use more visual communication tools
- Difference in terminologies
  - Understand the customer's terms
    - "He wants an elevator, we make only lifts"
  - Use Pictures and Diagrams



## Ambiguity From a Requirements Perspective - Pitfalls of the English language

- There are 5,790 languages in the world today
- More than one half of English speaking people did not grow up speaking it
- English has more words than any other language
- Estimates range from 490,500 words to nearly 2,000,000 words



## Ambiguity and Pitfalls

- Sloppiness

- From an airline safety booklet (found in the seat pocket)
  - "If you are sitting in an exit row and you cannot read this card or cannot see well enough to follow these instructions, please tell a crew member."

- Linguistic Ambiguity

- One half of two and two = ??



Copyright © Capgemini 2015. All Rights Reserved 9

## Ambiguity Checklist

- Dangling Else
- Ambiguity of reference
- Scope of action
- Omissions
  - Causes without effects
  - Missing effects
  - Effects without causes
  - complete omissions
  - Missing causes



- Ambiguous statements
  - Verbs, adverbs, adjectives
  - Variables, unnecessary
  - Aliases
- Random organization
  - Mixed causes and effects
  - Random case sequence

## Ambiguity Checklist

- Ambiguous Logical Operators
  - OR, AND
  - Implicit connectors
  - Compound operators
  - Negation
  - Scope of negation
  - Unnecessary negation
  - Double negation



- Built-in assumptions
  - Functional/environmental knowledge
- Unclear sequence
- Implicit case
- i.e. versus e.g.
- Temporal ambiguity
- Boundary ambiguity

## Ambiguity Checklist

### Dangling Else

Must be, will be, is one of, should be, could be, can be, shall

EXAMPLE:

"The loan type must be first or second."

Else?

An error condition?

### Ambiguity of Reference

EXAMPLE 1:

"Add Purchase-Amount to Account-Balance. This number must be positive."

EXAMPLE 2:

"Transaction 1 displays the customer's name and address.  
Transaction 2 displays the customer's account numbers.  
Transaction 3 displays the customer's account balances.  
Such transactions require the security code."



Copyright © Capgemini 2015. All Rights Reserved 12

## Ambiguity Checklist

### Omissions

- Causes without effects:  
"Codes 1 through 4 produce the message. It is also possible for the code to be a 5."
- Effects without causes:  
"This message sometimes appears."  
"It is sometimes necessary for the operator to re-initialize the field."
- Complete omissions  
A Blank page.  
Page unintentionally left blank.

### Missing causes:

EX.1 - "If you drive through a red light you will get a ticket."  
Missing - you must be caught doing it.

EX.2 - "If the number is 1, 3, 5, 7, 11, 13, 19,23, or 29 it is a prime number."  
Missing - 2, 17.

### Missing effects:

EX.1 - "If the account is overdrawn reject the check."  
Missing - notify the customer.



Copyright © Capgemini 2015. All Rights Reserved 13

## Ambiguity Checklist

- Ambiguous Logical Operators

**OR:**

If A or B then C.  
What people write:  
“A and B each produce C.”  
“A and B produce C.”

**AND:**

If A and B then C.  
What people write:  
“A and B produce C.”  
“A and B together are required to produce C.”

- Implicit Connectors:

Harry's “Going to the Party” Rules  
Rule 1: If either Sally or Sarah go, Harry will go.  
Rule 2: If Sally does not go with John, Harry will go.  
Rule 3: If Sarah does not go with Bob, Harry will go.

What happens if Sarah goes and Bob goes?

Rule 1: Says Harry will go.  
Rule 3: Implies Harry will not go.

- Confusing Compound Connectors:

Confusing compound operators:

“If A or B and C produce D.”  
If [A or B] and C produce D?  
OR  
If A or [B and C] produce D?



Copyright © Capgemini 2015. All Rights Reserved 14

## Ambiguity Checklist

- Ambiguous Statements

- Ambiguous verbs:

Calculate, update, produce, modify:  
"If it is month end calculate the interest earned."

- Ambiguous variables:

"If the interest amount is greater than \$100, send the customer the notice."

Interest accrued?  
Interest earned?  
Interest paid?  
Interest anticipated?

- Ambiguous adjectives:

"It is against the law to ride down the street on an *ugly* horse."      *Law in Wilbur, Washington*

- Ambiguous adverbs:

▪ "The delete transaction must be processed quickly."

▪ "Field A is usually positive."



Copyright © Capgemini 2015. All Rights Reserved 15

## Ambiguity Review

### ▪ Ambiguity Review Tips

- Be sensitive to the language issue by using simple, straightforward words
- Ambiguity reviews should eliminate instances of careless writing
- Ambiguity reviews by non-domain experts help eliminate assumed functional knowledge. Jargon should be avoided or at least defined in the glossary
- Acronyms must be defined in the data dictionary
- Do not write like you are being paid by the word

### ▪ Benefits of Ambiguity Reviews

- Timely feedback early in development life cycle
- Feedback leads to defect avoidance
- SA writing skills are improved



## Requirement Gathering - Skills Required

- Inter-Personal Skills
  - Effective communication
  - Probing, Right questioning !
    - Awareness
    - Listening
    - Compassion
- Engaging with the Customer
  - Expectations
  - Executive and User level
  - Organize Users and IT groups
  - Changes to Scope or terms of contract
  - Organization dynamics Skills for
  - Requirement Gathering
- Consulting Ability
  - Domain expertise
    - Business terms, Regulatory and statutory knowledge
  - Technology expertise
  - Ability to anticipate requirements
  - Pro-activeness, Problem Solving
  - Project Management
- Risk Management
  - Business, technology, operation
- Creativity, Innovation, Team work!!



Copyright © Capgemini 2015. All Rights Reserved. 17

## Tips to Requirement Gathering

- It is a team work!
  - Form the requirements elicitation team
  - Business & technical knowledge, experience
  - Gather knowledge on similar projects
- Training
  - Domain, Soft skills



Copyright © Capgemini 2015. All Rights Reserved. 18

## Tips to Requirement Gathering

- Know your Customer
  - Customer's profile and history
  - Market positioning, competitive edge/ differentiators
  - Business areas and core business domain
  - Business processes
  - Business risks
  - Confidentiality and sensitivity of business information
  - Regulatory and statutory knowledge
  - Customers' org structure, reporting structure of the key contacts-up
  - Organizational culture
  - Regional culture and language
  - Client Holidays
  - Our prior experience with the same customer



Copyright © Capgemini 2015. All Rights Reserved 19

## Tips to Requirement Gathering

- Identify stake holders and set the expectations
  - Identify single point contacts
  - Availability of business users
  - Ground work or preparation required from customer
  - Decision makers
- Identify participants
  - For interviews
  - Joint Application Development/Design [JAD] sessions
  - Brainstorming
  - Representatives from various groups
    - IT, Business, Users



Copyright © Capgemini 2015. All Rights Reserved 20

## Tips to Requirement Gathering

- General principles for requirements
  - Specify the problem – not the solution
  - Specify the system – not the project
  - Separate the formal and in-formal parts
  - Avoid repetition
  - Use the same lingo



Copyright © Capgemini 2015. All Rights Reserved 21

## Identification and Verification of Requirements

- Verify the gathered requirements with the customer
  - Can be done periodically through review meetings and status reports
  - Reverse walk through to eliminate bad or wrong requirements
- Summarize and present the requirements as understood by you to the user/stakeholder
  - Clarify any unclear requirements or conflicts
  - Clarify any unclear requirements or conflicts
- The requirements are presented “as you told” to the user



Copyright © Capgemini 2015. All Rights Reserved 22

Once the identification and verification of requirements is done. Approved SRS will be given to testing team. It is used for designing test cases.

Once SRS is received it is necessary to study those requirements and itemize those requirements in simple requirements.

## Summary

- In this lesson, you have learnt:
  - Requirements Gathering
  - Requirement Gathering Patterns
  - Challenges in Requirements
  - Ambiguity From a Requirements Perspective - Pitfalls of the English language
  - Ambiguity and Pitfalls
  - Ambiguity Checklist
  - Ambiguity Review
  - Requirement Gathering - Skills Required
  - Tips to Requirement Gathering
  - Identification and Verification of Requirements



## Review Question

- Question 1: Extreme approach strives to make product/application ASAP and generally requirements needs to be elaborated.
  - True/ False
- Question 2: Different terms used while stating requirements improves the understanding of requirement in the team.
  - True/ False
- Question 3: \_\_\_\_\_ approach emphasize on detailing the requirements in an incremental manner.
- Question 4: Probing and right questioning falls under \_\_\_\_\_ requirement gathering skills.



## Review Question

- Question 5: In which of the following requirement gathering patterns requirements are specified in detail and passes thru multiple reviews and sign-offs?
  - Option 1: Traditional approach
  - Option 2: Extreme approach
  - Option 3: Incremental approach
  - Option 4: None of the above



# **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)



Copyright © Capgemini 2015. All Rights Reserved 5

## Why do requirements change?

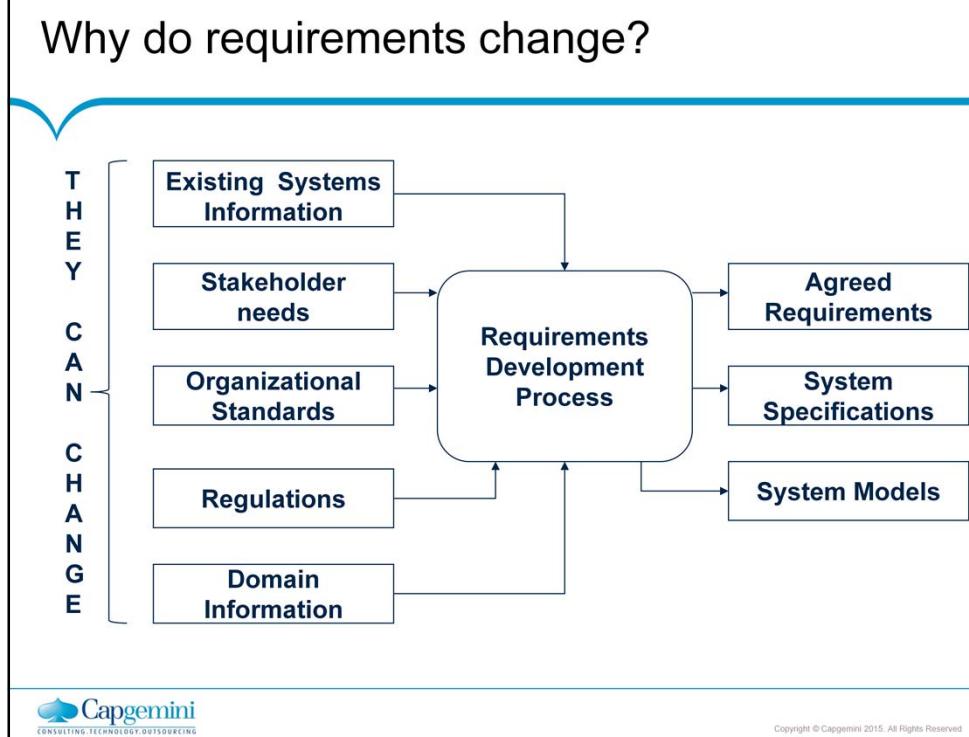
- Requirements change because :
  - Initial elicitation activities are imperfect
  - The priority of requirements from different viewpoints changes during the development process
  - Business needs evolve
  - Customers may specify requirements from a business perspective that conflict with end-user requirements
  - 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

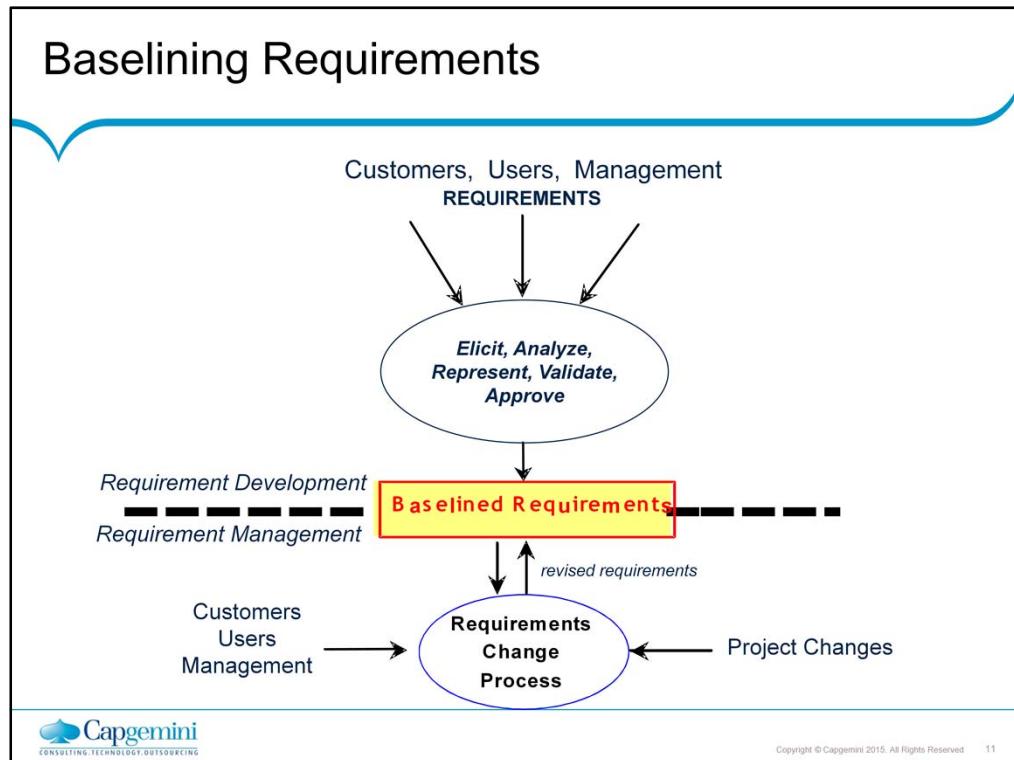


Copyright © Capgemini 2015. All Rights Reserved 10

## 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
<b>Requirements</b>					
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

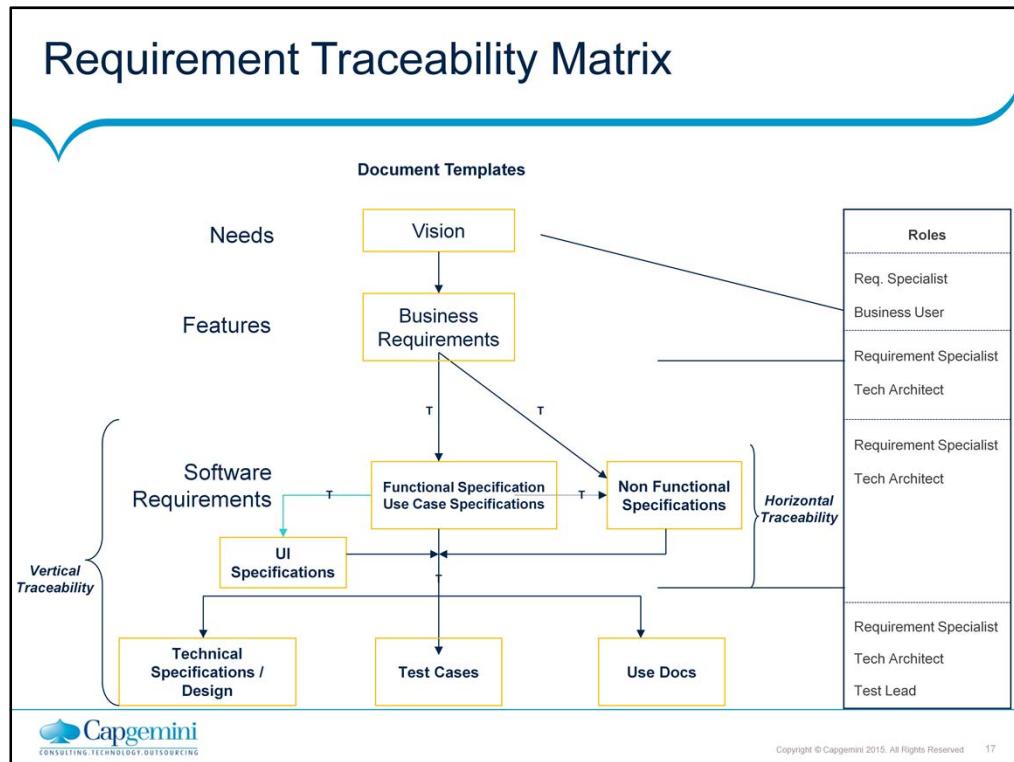

Capgemini  
CONSULTING TECHNOLOGY OUTSOURCING

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	<a href="http://www.FlyWithMe.com/Login.aspx">http://www.FlyWithMe.com/Login.aspx</a> 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	<a href="http://www.FlyWithMe.com/Login.aspx">http://www.FlyWithMe.com/Login.aspx</a> 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.

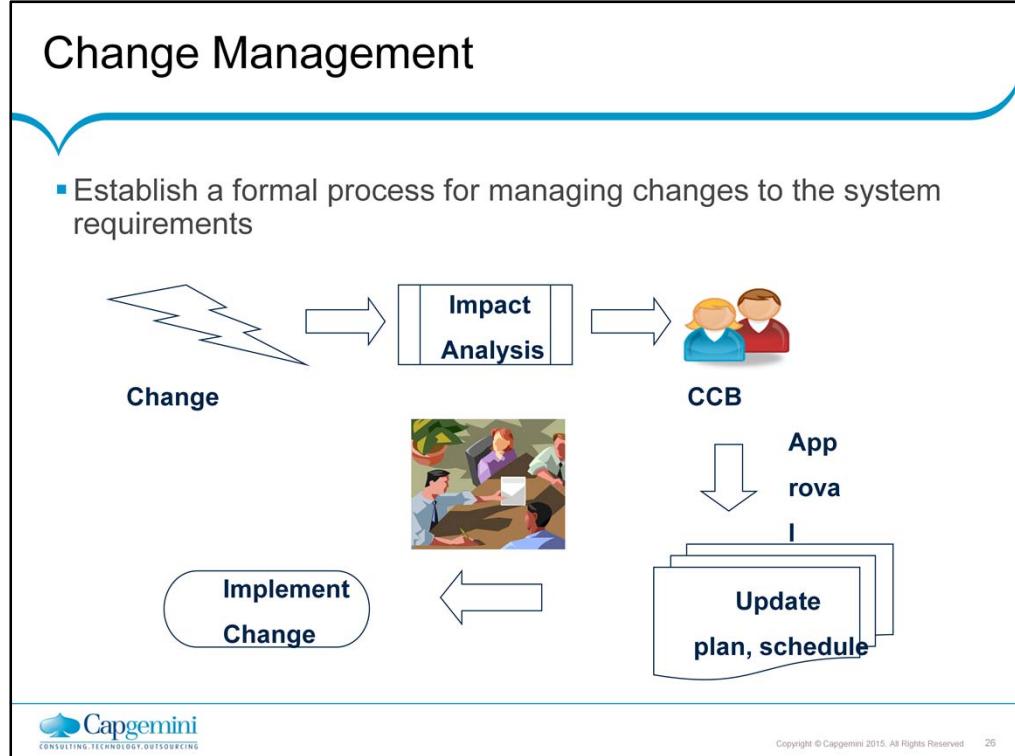


Copyright © Capgemini 2015. All Rights Reserved 25

### 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"><li>Minimize the scope creep</li><li>Increase requirements stability</li></ol>	<ol style="list-style-type: none"><li>Requirements Creep Index (RCI)</li><li>Requirements Stability Index (RSI)</li></ol>
	Elicitation	<ol style="list-style-type: none"><li>Increase requirements elicitation</li></ol>	<ol style="list-style-type: none"><li>Elicitation Efficiency (EEI)</li></ol>
Requirements Management	Query Resolution	<ol style="list-style-type: none"><li>Increase query resolution</li><li>Decrease time for closing a query</li></ol>	<ol style="list-style-type: none"><li>Query Resolution Index (QRI)</li><li>Average Query Turnaround Time (QTAT)</li></ol>



Copyright © Capgemini 2015. All Rights Reserved 35

## Requirement Metrics

Metrics	Overview	Calculation
<b>Requirements Creep Index (RCI)</b>	A measure of scope creep over gathered requirements	$((\text{Effort due to scope creep}) * 100) / (\text{Approved efforts in person days as approved by customer})$
<b>Requirements Stability Index (RSI)</b>	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.
<b>Elicitation Efficiency (EEI)</b>	A measure of the efficiency of elicitation process	Use of a template for calculating EEI
<b>Query Resolution Index (QRI)</b>		Total number of Queries Resolved / Total number of Queries
<b>Average Query Turnaround Time (QTAT)</b>	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



# Requirements Validation and Functional Decomposition Lab Book

## Document Revision History

Date	Revision No.	Author	Summary of Changes
12/8/09	1	Priya Rane	Revamp
21/6/2011	1.1	Selvalakshmi	Material Revamp
18/6/15	2.0	Alphy Thomson & Shilpa Bhosle	Material Revamp

## Table of Contents

<i>Document Revision History</i> .....	2
<i>Table of Contents</i> .....	3
<i>Problem Statement/ Case Study (If applicable)</i> .....	5
<i>Lab 1. Requirements Validation and Functional Decomposition</i> .....	9
a.     Login function .....	13
b.     Add Exam Details function .....	14
c.     Update Exam Details function .....	16
d.     Delete Exam Details function .....	17
a.     Login function .....	19
b.     Make Payment function .....	21
c.     View Application function.....	22
d.     Print Receipt Function .....	23
<b>1.4: External Interface Requirement - User Interfaces</b> .....	24
<i>Home Page</i> .....	24
<i>List of Exam and Their details</i> .....	25
<i>Admin Home Page</i> .....	26
<i>Add an exam Page</i> .....	27
<i>Update an exam Page</i> .....	29
<i>Update selected Exam Page</i> .....	30
<i>Delete an exam Page</i> .....	33
<i>New Registration Page</i> .....	34
<i>Applicant Home Page</i> .....	37
<i>Payment Page</i> .....	38
<i>Receipt Page</i> .....	39
<i>View Application Page</i> .....	40
<i>Lab 2. Identify ambiguities in the requirements</i> .....	42
1. <i>Scope QWERTYsoft - Requirements Document</i> .....	42
1.1 <i>Terms, Acronyms, and Abbreviations</i> .....	42
2. <i>Required release date</i> .....	43
3. <i>Description of requirements</i> .....	43
3.1 <i>General technical requirements</i> .....	43
4. <i>Welcome</i> .....	43
5. <i>Payment</i> .....	43
6. <i>Internet Browser</i> .....	43

7. <i>Performance</i> .....	44
8. <i>Localization</i> .....	44
9. <i>Content Control</i> .....	44
10. <i>Session Termination</i> .....	44
11. <i>Confidentiality</i> .....	44
12. <i>Administration</i> .....	44
13. <i>View Kiosks</i> .....	45
14. <i>View Users</i> .....	45
15. <i>Modify User</i> .....	45
16. <i>Terminate User</i> .....	45

## Problem Statement/ Case Study (If applicable)

### Equipment Tracking System

#### About Application

##### Background

1. System - Equipments tracking system.
2. Most of the functionality available on Fat client and some on Web.
3. Module under test – Manage Equipments
4. Flow of the system.  
Purchase equipment → Manage Equipment → Track financials→Retire equipment
5. Manage equipment means – Install, Update and Move equipment.
6. There are different access rights for different users – (Browse and Edit.)

Note – The actual field names on the GUI screen can be different than the listed in the requirement.

#### Assumptions

1. Equipment entry is already in the system as Purchase phase is over.
2. Equipment 'Use status' can be 'In Use' if it is installed and assigned to some user/Department or it can be 'In stock' if it is not put in Use.

#### Requirements

1. Authorized users and systems must have the ability to record updates to equipment records
  - 1.1. The system must enforce requirements of the equipment Type properties while updation.
  - 1.2. Only valid Locations can be assigned to an equipment record when completing an update transaction. End Date has not expired
  - 1.3. Only valid Users can be assigned to an equipment record when completing an update transaction. End Date has not expired
2. The system must provide the ability to restrict which users may update equipment records.
  - 2.1. The authorized users are: Inventory personnel, equipment auditors, service personnel, maintenance personal and Equipment Tracking personnel
3. The system must provide the user with the capability to query for equipment records based on a predefined list of available data elements.
  - 3.1. Users must have the ability to query for an equipment by "Equipment Tag"
  - 3.2. The system must manipulate the barcode Equipment Tag for query and storage consistency
    - 3.2.1. Equipment Tag must be 10 characters
    - 3.2.2. Equipment Tag must be all numeric
    - 3.2.3. Equipment Tag will remove any blanks or dashes
    - 3.2.4. The text 'AD' will be removed from Equipment Tag if found in position 1 and 2 of

- the character string
- 3.2.5. Equipment Tag will be padded with leading zeros if length is less than 10 characters
- 3.3. Users must have the ability to query for equipments by "Seq Number"
  - 3.3.1. The system must manipulate the Seq number input for query and storage consistency
  - 3.3.2. The system must use the most recent rules contained within the authorized document:
- 3.4. Users must have the ability to query for an equipment by "Machine Id"
- 3.5. Users must have the ability to query for equipments by "User Id"
- 3.6. Users must have the ability to query for equipments by "Location"
  - 3.6.1. Users must select a valid Location from a list.
- 4. System should return a list of equipments matching the query criterion
  - 4.1. The system must provide the ability to view the Equipment Tag for each equipment record returned in the list.
  - 4.2. The system must provide the ability to view the Quantity for each equipment record returned in the list..
  - 4.3. The system must provide the ability to view the Seq. Number for each equipment record returned in the list.
  - 4.4. The system must provide the ability to view the User Id for each equipment record returned in the list.
  - 4.5. The system must provide the ability to view the Location for each equipment record returned in the list.
  - 4.6. The system must provide the ability to view the Equipment type for each equipment record returned in the list.
- 5. The system must provide the user with the ability to manually update specified data for a single equipment record.
  - 5.1. Users are restricted to modifying equipment records only in within their state
  - 5.2. Only comments can be updated for retired equipments
  - 5.3. Any Unassociated equipments identified as "Spare Part" must be 'In Stock'
- 6. The system must provide the user with the ability to manually update specified data for multiple equipment records simultaneously.
- 7. Following is the list of updatable fields depending on the use transactions and equipment status
  - 7.1. Purchase Method - Required
  - 7.2. Seq Number - Required based on the equipment type.
  - 7.3. User ID - If allocated to user
  - 7.4. Department ID - If allocated to department
  - 7.5. Use Status – Required. If stock equipment then default to In-Stock
  - 7.6. Cost Center –Required. If Use status is In-Stock default with Stock Location's cost center ID
  - 7.7. Install Date Required if Use Status is 'In Use'
    - Default to actual Receipt Date if receiving done through automatic interface.
    - Default to Current Date if equipment is added into system manually.
  - 7.8. Location- Required

- 7.9. Audit Indicator (Yes/No)
- 7.10. Audit Date - Default the Audit Date to the Receipt Date
- 7.11. comments
- 7.12. Stock Location - Required if use status is In-Stock
- 8. The system must provide the ability to update specified data in an equipment record from external systems.
- 9. System should allow printing the equipment's basic information using 'Print Label' functionality.
  - 9.1. Label should include
    - 9.1.1. Equipment code
    - 9.1.2. User ID
    - 9.1.3. Location ID
  - 9.2. System should receive automatic updates from the 'CompTrak' system for all the computer related fields on a regular basis. ('CompTrak' is the network-based system that tracks all the computers in the network and collects data physically for all the computers. So this system is the Primary data holder for the computer specific fields)
    - 9.2.1. Computer Name
    - 9.2.2. Disk Capacity
    - 9.2.3. Total installed memory
    - 9.2.4. Network Card number
    - 9.2.5. Network Card Manufacturer
    - 9.2.6. Free space on 'C' Drive
    - 9.2.7. Operating system
    - 9.2.8. OS version
  - 9.3. Updations are done only if the last scan date for a stored in the 'Equipment tracking system' is lesser than the 'Last Scan date' stored in the 'CompTrak' system
  - 9.4. Errors would be generated if
    - 9.4.1. The equipment tag is not found in 'Equipment tracking system'
    - 9.4.2. Last scan date for a stored in the 'Equipment tracking system' is greater than the 'Last Scan date' stored in the 'CompTrak' system
- 10. System should supply the detailed information in the comma separated flat file format to the statistical department
  - 10.1. This should be automatic process run on last day every month
  - 10.2. It should include those equipments that are received in current month
  - 10.3. It should exclude any 'Retired' equipment even if they are purchased this month.
  - 10.4. Following fields should be present
    - 10.4.1. Use status
    - 10.4.2. Type of equipment
    - 10.4.3. Location
    - 10.4.4. Total quantity
    - 10.4.5. Purchase Date
  - 10.5. Apart from flat file, an excel file should also be created which will contain the same data (except purchase date) but grouped by Use status.
    - 10.5.1. Sorting should be by Use status, Type and Location(All Ascending)

- 10.5.2. Quantity Subtotals should be available for each change in Use status and Type of equipment.
- 10.5.3. The detailed format for this Excel report is available in the <Some logical information>(Assume that this information is available)

## Lab 1. Requirements Validation and Functional Decomposition

<b>Goals</b>	<ul style="list-style-type: none"><li>• Understand the process of requirements itemization.</li><li>• Learn to apply basic techniques of requirement itemization to understand the requirements better.</li><li>• Learn to prepare requirement itemization document.</li></ul>
<b>Time</b>	180 minutes

# FUNCTIONAL REQUIREMENTS SPECIFICATION FOR **Banking Exam Portal** Project Code: UI026

**Version 2.0**

## 1.1: System Overall Description

The web site of Banking Exam Portal should be accessible to anyone who can log onto the World Wide Web using a standard browser that is version 8.0 or higher. The inventory of this banking exam portal should be maintained in a relational database, such as Oracle. A combination of standard HTML and a scripting language for web application, such as Cold Fusion of MS Server Pages, should be considered in developing the web page interfaces.

The application uses Oracle 9i as the backend database.

All internal and external exceptions occurring in the system are handled through a custom exception handling mechanism using an application specific exception class. All error messages are conveyed to the users through the standard exception handling mechanism provided. The exceptions are sent to web pages and mapped to friendly messages on the web pages. Fatal error messages are also logged to an application specific log file with the appropriate details.

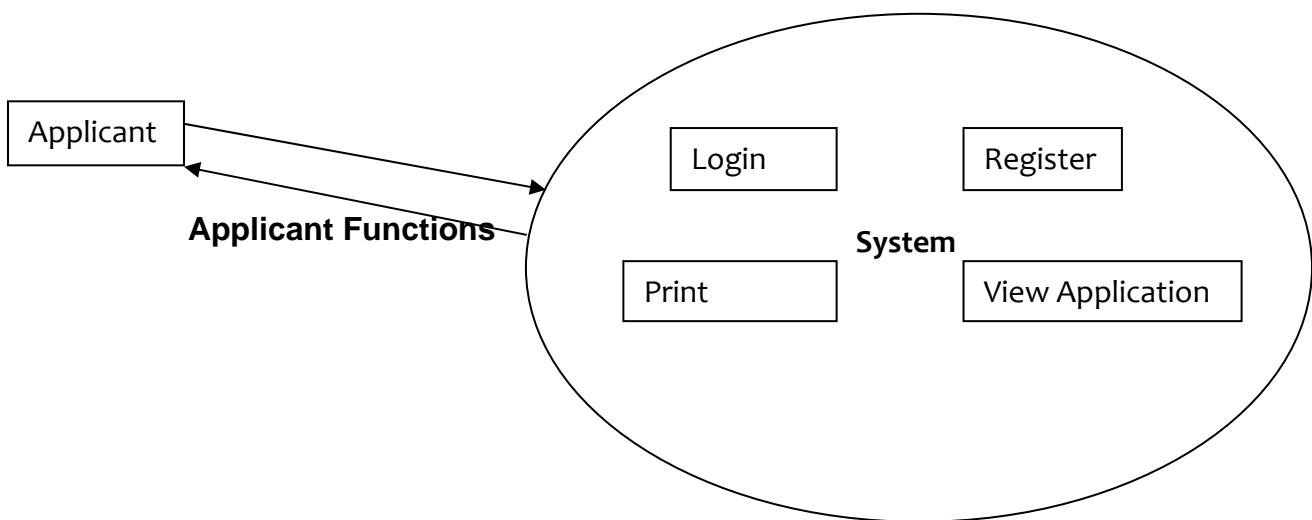
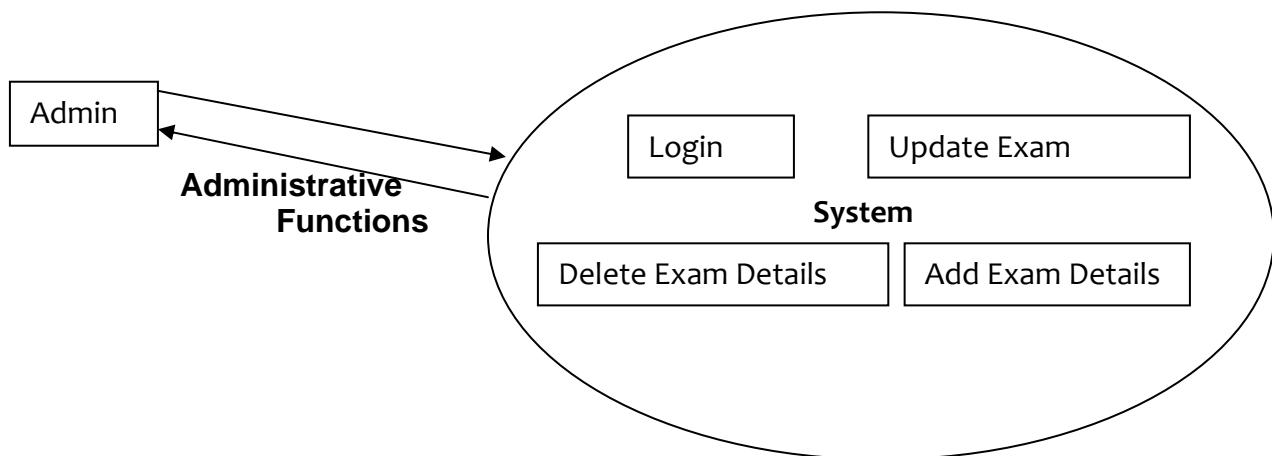
The following table provides details of the environment for end-user

Software Specification
<ul style="list-style-type: none"><li>• Windows Vista, Windows 7, Windows 8, Windows 8.1</li><li>• Internet Explorer or any standard browser 8.0 and above</li><li>• Oracle 9i</li></ul>

## 1.2: System Description

This exam portal can be accessed by two different users: Admin and Applicant. The operations performed by both of them are shown in the Data Flow Diagrams below. The portal can be accessed by entering the name of the website ([www.bankersjob.com/exam](http://www.bankersjob.com/exam)) in the browser. The admin can directly login to the application using the credentials given to him. The applicant can login only after applying for an exam. To apply for an exam the applicant needs to register.

### Data Flow Diagrams of Banking Exam Portal



### 1.3: Product Functions

Function No.	Function Name	Sub-Function No.	Sub-Function(s)
1.0	Administrative Functions	1.0.1	Login
		1.0.2	Add Exam Details
		1.0.3	Update Exam Details
		1.0.4	Delete Exam
2.0	Applicant Functions	2.0.1	Register
		2.0.2	Login
		2.0.3	Make Payment
		2.0.4	View Application
		2.0.5	Print Receipt
3.0	Security Functions	3.0.1	Accessibility of each page
4.0	Exception Handling Functions	4.0.1	

## 1. Administrative Functions

### a. Login function

Function No.	Function Name	Description
1.0.1	Login	<ul style="list-style-type: none"> <li>▪ The admin enters his/her user-id and password and clicks the login button</li> <li>▪ The system validates the user-id and password combination with the database.</li> <li>▪ If the authentication is unsuccessful, a login failure message is displayed.</li> <li>▪ On successful login, the admin home page is displayed.</li> <li>▪ On Successful login, the user-id and role fields are stored in the HTTP session for future use during that session. The user session expires after a default inactivity interval of 20 minutes.</li> </ul>

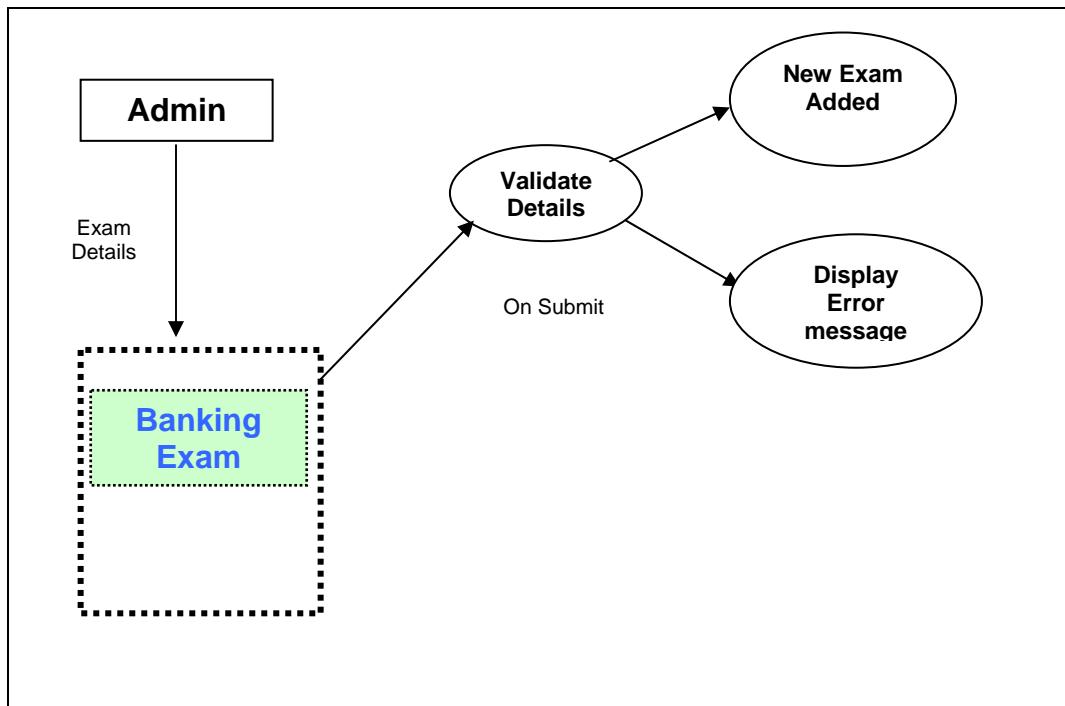
```

graph TD
    Admin[Admin] -- "User Id" --> BEP[Banking Exam Portal]
    Admin -- "Password" --> BEP
    BEP --> VU((Validate User))
    VU --> DAHP([Display Admin Home Page])
    VU --> DEM([Display Error message])
  
```

The flowchart illustrates the login process. An external actor 'Admin' provides 'User Id' and 'Password' to a boundary object 'Banking Exam Portal'. This portal then interacts with a process 'Validate User', which leads to either 'Display Admin Home Page' or 'Display Error message'.

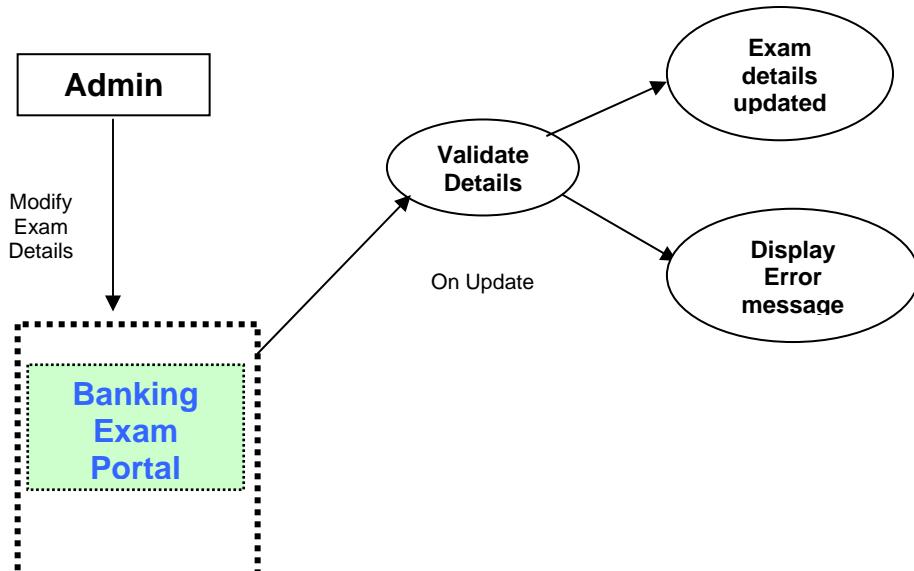
### b. Add Exam Details function

Function No.	Function Name	Description
1.0.2	<b>Add Exam Details</b>	<ul style="list-style-type: none"> <li>▪ When the Bank announces a particular exam, the admin adds the exam details to the list of exams.</li> <li>▪ The exam added should have a unique name. Once submitted the exam name cannot be changed.</li> <li>▪ The commencement of online registration date should be mentioned and the closure date for registration would be 30 days from the commencement date.</li> <li>▪ The date of the exam should be at least 60 days ahead of the commencement date.</li> <li>▪ The examination fees needs to be mentioned along with the exam details.</li> <li>▪ On clicking of the submit button all the details should be validated.</li> <li>▪ If any of the fields are not entered appropriate error messages needs to be shown</li> <li>▪ Also if any of the fields are incorrectly entered suitable messages should get displayed</li> <li>▪ Once the exam is added, the same should be displayed in the home page and the Applicant can apply for the exam.</li> </ul>



### c. Update Exam Details function

Function No.	Function Name	Description
1.0.3	<b>Update Exam Details</b>	<ul style="list-style-type: none"> <li>▪ When some existing exam details needs to be updated ,the admin does so through the update exam details functionality</li> <li>▪ Everything except the exam name can be changed for a particular exam</li> <li>▪ The admin selects the particular exam to be updated and clicks on the update button</li> <li>▪ On the following page the exam details appear as editable fields.</li> <li>▪ Once the modification is done the updated exam details are populated in the home page</li> <li>▪ If any of the fields are not entered appropriate error messages needs to be shown</li> <li>▪ Also if any of the fields are incorrectly entered suitable messages should get displayed</li> </ul>

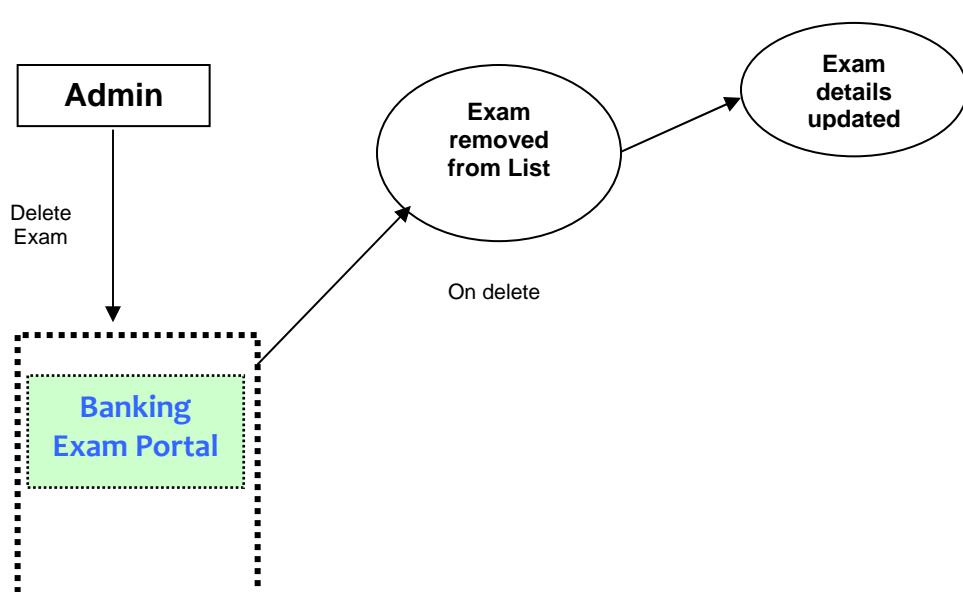


```

graph TD
    Admin[Admin] -- "Modify Exam Details" --> BEP[Banking Exam Portal]
    BEP --> VD{Validate Details}
    VD -- "On Update" --> EDU([Exam details updated])
    VD -- "On Update" --> DEM([Display Error message])
  
```

#### d. Delete Exam Details function

Function No.	Function Name	Description
1.0.4	Delete Exam	<ul style="list-style-type: none"> <li>When some existing exam details needs to be deleted ,the admin does so through the delete exam functionality</li> <li>The admin selects the particular exam to be deleted and clicks on the delete button</li> <li>Once it is deleted the exam name is removed from the list of the exams available</li> </ul>

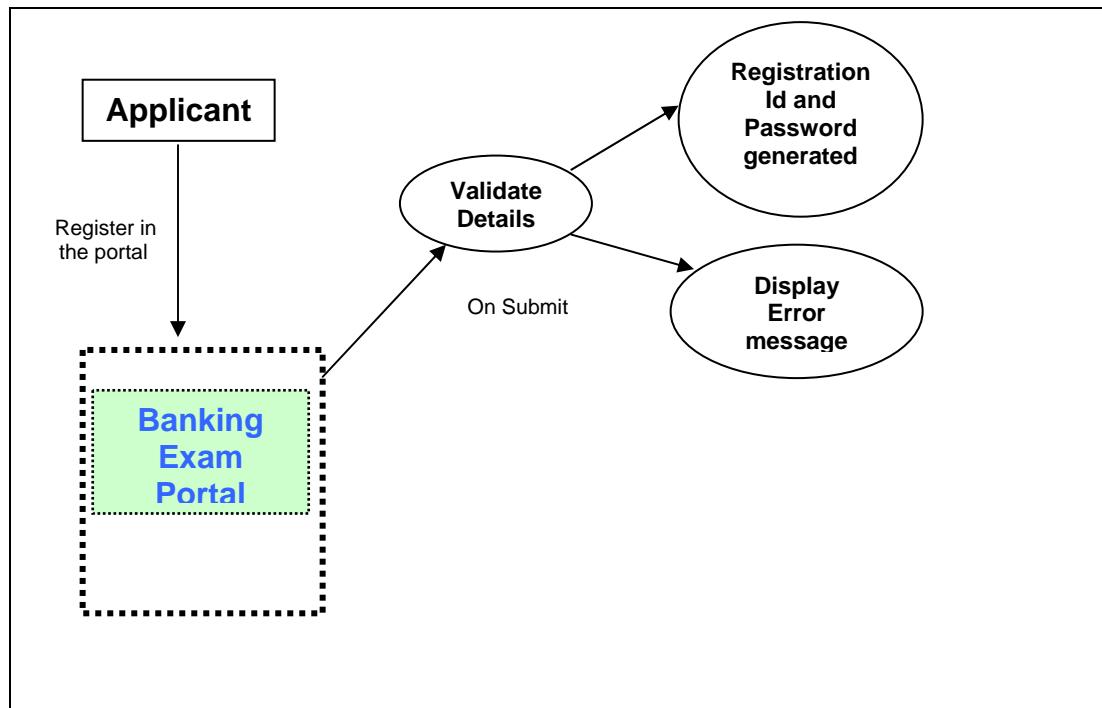
```

graph TD
    Admin[Admin] -- "Delete Exam" --> BankingExamPortal[Banking Exam Portal]
    BankingExamPortal -- "On delete" --> ExamRemoved[Exam removed from List]
    ExamRemoved --> ExamDetailsUpdated[Exam details updated]
  
```

## 2. Applicant's Functions:

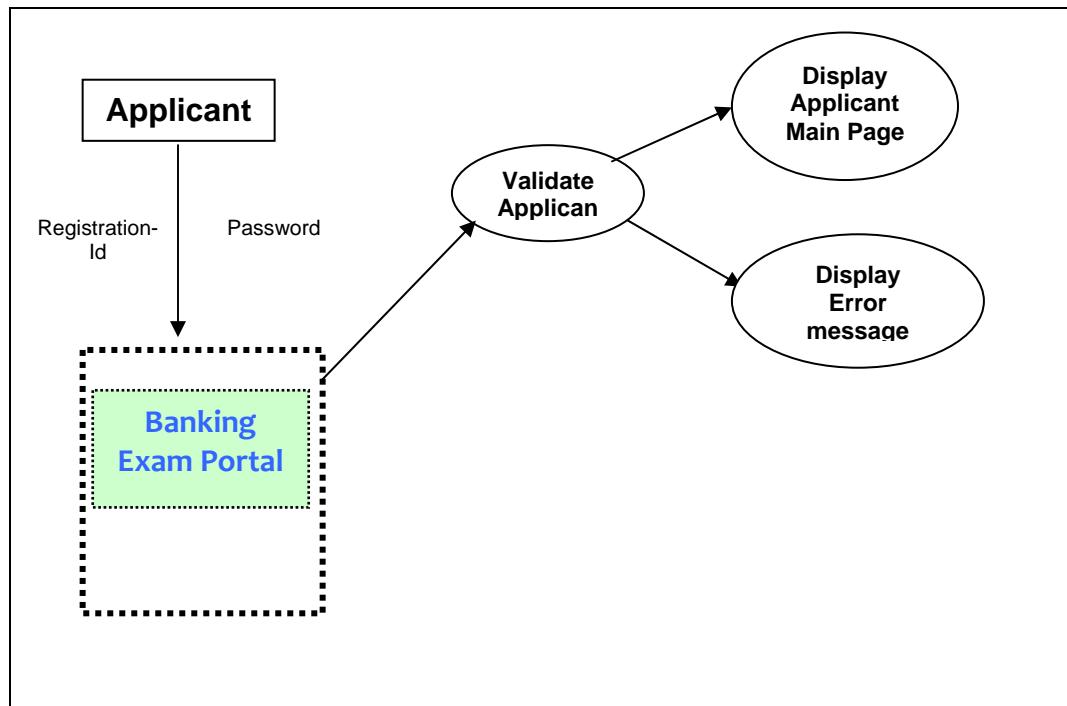
### a. Register Function

Function No.	Function Name	Description
2.0.1	Register Function	<ul style="list-style-type: none"> <li>▪ When a new applicant visits the portal and wishes to apply for an exam, he is required to register with the site.</li> <li>▪ In the registration section the full name gets populated from the first and last name entered.</li> <li>▪ Other personal details like mobile number, email id, DOB, Gender, Marital Status are mentioned.</li> <li>▪ Educational qualification details like degree, subject, year of passing, percentage of marks are to be collected.</li> <li>▪ Photo and signature should be uploaded by the applicant as image files.</li> <li>▪ On clicking on the Save &amp; Submit button the data entered is validated.</li> <li>▪ In case of any of the fields are not entered appropriate error messages needs to be shown</li> <li>▪ Also if any of the fields are incorrectly entered suitable messages should get displayed</li> <li>▪ If the values are validated then it should generate and display the registration id, password and the same are sent to the registered mobile no.</li> </ul>



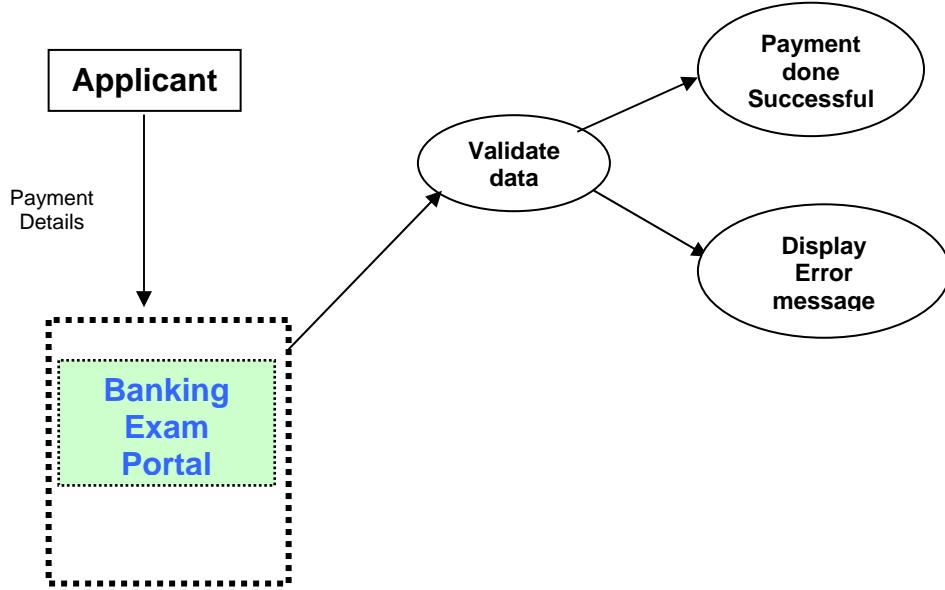
### a. Login function

Function No.	Function Name	Description
2.0.3	Login	<ul style="list-style-type: none"> <li>▪ The applicant can login into the portal with the registration id and password received after successful registration.</li> <li>▪ This is done through the Login function available through the 'Existing Candidate Login' option</li> <li>▪ The system validates the registration-id/password combination with the database.</li> <li>▪ If the authentication is unsuccessful, a login failure message is displayed.</li> <li>▪ On successful login, the Main page for Applicant is displayed.</li> <li>▪ On Successful login, the registration-id and role fields are stored in the HTTP session for future use during that session. The applicant session expires after a default inactivity interval of 20 minutes.</li> <li>▪ The Main page contains options of viewing the application, making payment and printing the receipt of the payment done.</li> </ul>



### b. Make Payment function

Function No.	Function Name	Description
2.0.2	Make Payment Function	<ul style="list-style-type: none"> <li>▪ Once the user has been successfully registered on the website he/she can login to the banking exam portal to make payment for the exam that he/she has registered for.</li> <li>▪ The applicant makes use of the Make Payment Function to do this</li> <li>▪ The portal supports only debit card payment option</li> <li>▪ In the payment page the details of bank, debit card details like card no, validity date, card holder's name, PIN are asked for.</li> <li>▪ On successfully entering these details and clicking on the Make Payment button the payment is done successfully followed by a message being displayed.</li> </ul>



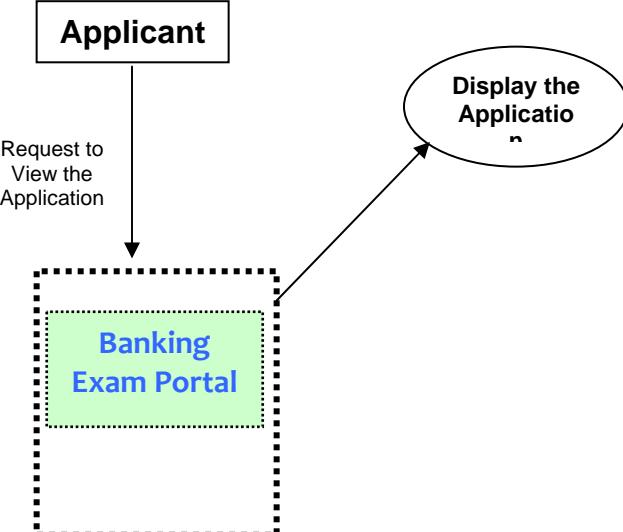
```

graph TD
    Applicant[Applicant] -- "Payment Details" --> BankingExamPortal[Banking Exam Portal]
    BankingExamPortal --> ValidateData((Validate data))
    ValidateData --> PaymentSuccessful((Payment done Successful))
    ValidateData --> DisplayError((Display Error message))
  
```

The flowchart illustrates the process for making a payment. It starts with an external entity, 'Applicant', which sends 'Payment Details' to the 'Banking Exam Portal'. Inside the portal, the data is validated. If successful, the process ends with a message of 'Payment done Successful'. If validation fails, an 'Error message' is displayed.

### c. View Application function

Function No.	Function Name	Description
2.0.4	View Application Function	<ul style="list-style-type: none"> <li>▪ The applicant can view the details of his application using the View Application function.</li> <li>▪ On clicking the view application option the report containing the details of the application is displayed.</li> </ul>



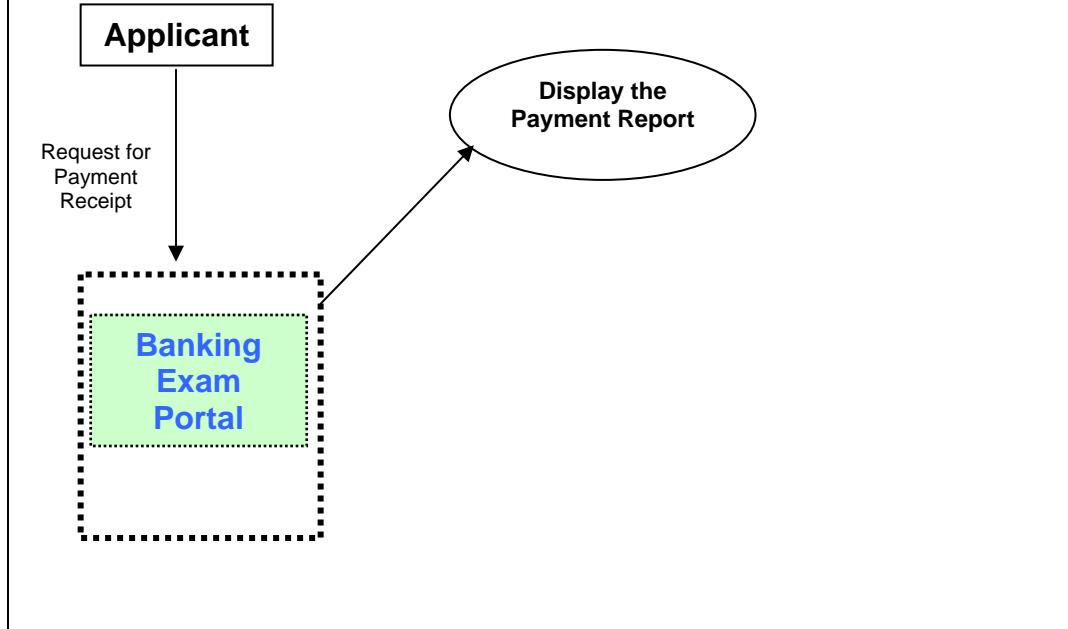
```

sequenceDiagram
    participant Applicant
    participant BEP [Banking Exam Portal]
    Applicant->>BEP: Request to View the Application
    activate BEP
    BEP->>Applicant: Display the Application
    deactivate BEP
  
```

The diagram illustrates a sequence of interactions. An external entity, 'Applicant', sends a 'Request to View the Application' message to a local object, 'Banking Exam Portal'. This object then returns a 'Display the Application' message back to the 'Applicant'.

#### d. Print Receipt Function

Function No.	Function Name	Description
2.0.5	Print Receipt Function	<ul style="list-style-type: none"> <li>The applicant can print the payment details of his application using the Print Receipt function.</li> <li>On clicking the Print Receipt option the report containing the details of the payment is displayed.</li> </ul>



In addition to the above functionality system also provides following features.

#### 1. Security

- a. Each page in the Banking Exam Portal web application checks for a valid user login by checking the value of a session variable. This scheme ensures that users cannot directly access any web page of the application by simply entering the URL of that page on the address bar of the browser.

#### 2. Exception Handling

- a. All exceptions occurring due to reasons external to the Banking Exam Portal system e.g. database/network related exceptions are trapped and appropriate error messages are displayed to the users.

## 1.4: External Interface Requirement - User Interfaces

### Home Page

Banking Exam Portal		
Welcome to the Banking Exam Portal!!		
<u>List of Exams on the Portal</u>	<p>Admin Login</p> <p>User Name: <input type="text"/></p> <p>Password : <input type="password"/></p> <p><input type="button" value="Submit"/></p>	<p>New Candidate? <a href="#">Click here to register</a></p> <p>Existing Candidate Login</p> <p>Registration No: <input type="text"/></p> <p>Password: <input type="password"/></p> <p><input type="button" value="Submit"/></p>

Purpose	:	Main Home Page
Functionality	:	<ul style="list-style-type: none"> <li>The registered users can login into the system.</li> <li>The list of exams can be viewed</li> <li>The user can register on the banking exam portal</li> </ul>

Control Label	Control Type/ Length/ Multi select	Description/ Business Rule	Default Value	Validation
List of Exams on the Portal	Hyperlink	Users can click to view the list of exams offered	-	It should navigate user to the appropriate web page containing list of exams
User Name (Admin)	Text Box	Specify the admin user name	-	Username entered is validated with the data in the database
Password (Admin)	Password Box	Specify the admin password	-	Password entered is validated with the data in the database
Submit (Admin)	Button	On click validates the user credentials and based on the validation	-	-

		result allows or disallows Admin from logging to the website		
Click Here	Hyperlink	On click new users can register to the website	-	It should navigate the new user to the <b>Register</b> web page
Registration No	Text Box	Specify the registration number	-	Registration No entered is validated with the data in the database
Password	Password Box	Specify the applicant's password	-	Password entered is validated with the data in the database
Submit	Button	On click validates the user credentials and based on the validation result allows or disallows user from logging to the website	-	

### List of Exam and Their details

Banking Exam Portal																	
<a href="#">Back</a>	List of Exams and their details	<a href="#">LogOut</a>															
<table border="1"> <thead> <tr> <th>Exam Code</th> <th>Exam Name</th> <th>Exam Date</th> </tr> </thead> <tbody> <tr> <td>AXISPOO1</td> <td>AXIS Probationary Officer's Exam 2015</td> <td>29/5/2015</td> </tr> <tr> <td>CANPOO1</td> <td>CANARA Probationary Officer's Exam 2015</td> <td>1/6/2015</td> </tr> <tr> <td>SBICL001</td> <td>SBI Clerk Exam 2015</td> <td>28/5/2015</td> </tr> <tr> <td>RBICL001</td> <td>RBI Clerk Exam 2015</td> <td>22/4/2015</td> </tr> </tbody> </table>			Exam Code	Exam Name	Exam Date	AXISPOO1	AXIS Probationary Officer's Exam 2015	29/5/2015	CANPOO1	CANARA Probationary Officer's Exam 2015	1/6/2015	SBICL001	SBI Clerk Exam 2015	28/5/2015	RBICL001	RBI Clerk Exam 2015	22/4/2015
Exam Code	Exam Name	Exam Date															
AXISPOO1	AXIS Probationary Officer's Exam 2015	29/5/2015															
CANPOO1	CANARA Probationary Officer's Exam 2015	1/6/2015															
SBICL001	SBI Clerk Exam 2015	28/5/2015															
RBICL001	RBI Clerk Exam 2015	22/4/2015															
<input type="button" value="OK"/>																	

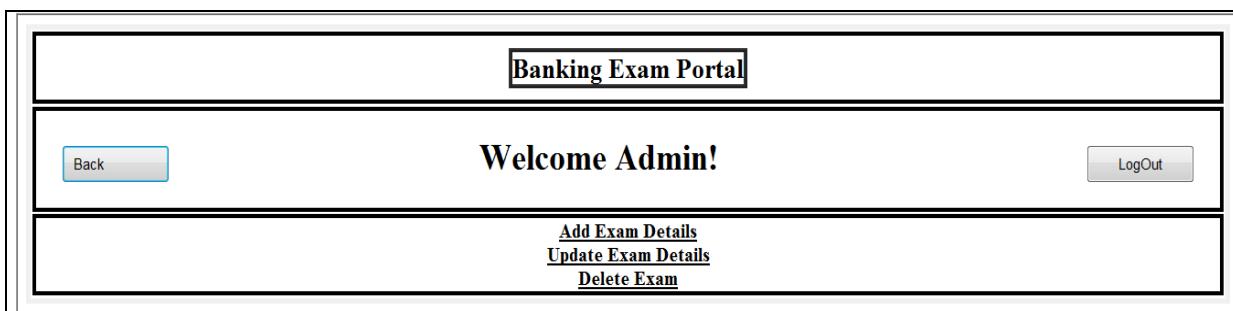
Purpose	:	List of Exam and Their details
Functionality	:	<ul style="list-style-type: none"> <li>This page gives the list of exams and their details</li> </ul>

Role	Details / Business Rule / Constraints
Applicant	These users can invoke this functionality.

Control Label	Control Type/ Length/	Description/ Business Rule	Default Value	Validation

	<b>Multi select</b>			
Back	Button	On click redirects to the main home page of the portal	-	-
LogOut	Button	On click logs out the admin from the portal	-	-
OK	Button	On click goes back to the home page	-	

### Admin Home Page



Purpose	:	Admin Home Page
Functionality	:	<ul style="list-style-type: none"> <li>The admin's functionality of adding, updating and deleting exam can be accessed from this page</li> </ul>

<b>Role</b>	<b>Details / Business Rule / Constraints</b>
Administrator	These users can invoke this functionality.

<b>Control Label</b>	<b>Control Type/ Length/ Multi select</b>	<b>Description/ Business Rule</b>	<b>Default Value</b>	<b>Validation</b>
Back	Button	On click redirects to the main home page of the portal	-	-
LogOut	Button	On click logs out the admin from the portal	-	-
Add Exam Details	Hyperlink	Admin can click to add a new exam and its details	-	It should navigate the Admin to the <b>Add Exam Details</b> web page
Update Exam Details	Hyperlink	Admin can click to update an exam's details	-	It should navigate

				the Admin to the <b>Update Exam Details</b> web page
Delete Exam	Hyperlink	Admin can click to delete an exam	-	It should navigate the Admin to the <b>Delete Exam</b> web page

### Add an exam Page

Banking Exam Portal	
<a href="#">Back</a>	<b>Add an Exam</b>
<a href="#">LogOut</a>	
Exam Name	<input type="text"/>
Exam Code	<input type="text"/>
Bank Name	<input type="text"/>
Commencement of Online Registration	<input type="button" value="Select Date"/>
Closure of Online Application	<input type="button" value="Select Date"/>
Exam Date	<input type="button" value="Select Date"/>
Description about the exam	<input type="text"/>
Examination Fees	<input type="text"/>
<input type="button" value="Submit"/> <input type="button" value="Reset"/> <input type="button" value="Cancel"/>	

Purpose	:	Add an Exam Page
Functionality	:	<ul style="list-style-type: none"> <li>A new exam and its details can be added using this page</li> </ul>

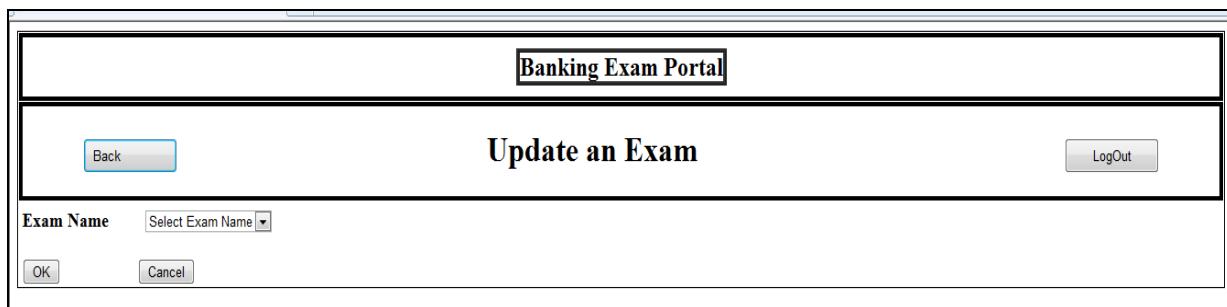
Role	Details / Business Rule / Constraints
Administrator	These users can invoke this functionality.

Control Label	Control Type/ Length/ Multi select	Description/ Business Rule	Default Value	Validation
Back	Button	On click redirects to the home page of the admin	-	-
LogOut	Button	On click logs out the admin from the portal	-	-

Exam Name	Text Box	Exam name for the exam can be given here	-	Should be a unique name
Exam Code	Auto Generated Text	On giving the exam name an exam code is generated by the system	-	A unique value should be generated after the exam name is entered
Bank Name	Text Box	Bank name can be given here	-	Should accept valid bank name
Commencement of Online Registration	Calendar Control	Date can be selected for the commencement of the online registration	-	Calendar should get populated while clicking on the control. It should be a future date.
Closure of Online Application	Calendar Control	Date can be selected for the commencement of the online application	-	Calendar should get populated while clicking on the control. Should be 30 days from the commencement date.
Exam Date	Calendar Control	Date can be selected for the exam	-	Calendar should get populated while clicking on the control. Should be more than 60 days from the date of commencement date.
Description about	Text Box	Description about the	-	It can

exam the exam		exam can be given here		accept characters and numeric values.
Examination Fees	Text Box	The fee amount can be entered here	-	It can accept only numeric values.
Submit	Button	On click it submits all the details	-	-
Reset	Button	On click it resets the details	-	-
Cancel	Button	On click cancels the operation and closes the web page	-	-

## Update an exam Page



The screenshot shows a web-based application interface for updating an exam. At the top, there's a header bar with the text "Banking Exam Portal". Below it is a main content area with a title "Update an Exam". On the left side of this area, there's a button labeled "Back". On the right side, there's a button labeled "Logout". In the center, there's a dropdown menu labeled "Select Exam Name". At the bottom of the content area, there are two buttons: "OK" and "Cancel".

Purpose	:	Select an exam to update
Functionality	:	<ul style="list-style-type: none"> <li>The exam to be updated is selected from this page</li> </ul>

Role	Details / Business Rule / Constraints
Administrator	These users can invoke this functionality.

Control Label	Control Type/ Length/ Multi select	Description/ Business Rule	Default Value	Validation
Back	Button	On click redirects to the home page of the admin	-	-
LogOut	Button	On click logs out the	-	-

Exam Name	Drop Down Box	admin from the portal	-	All the exams available should be present in the list
Ok	Button	On click ,redirects to the next page	-	-

## Update selected Exam Page

**Banking Exam Portal**

**Update Selected Exam**

<a href="#">Back</a>	<a href="#">LogOut</a>
<p><b>Exam Name:</b> Probationary Officers Exam</p> <p><b>Exam Code:</b> AXISPO01</p> <p><b>Bank Name:</b> Axis Bank</p> <p><b>Commencement of Online Registration:</b> 19/2/2015</p> <p><b>Closure of Online Application:</b> 19/3/2015</p> <p><b>Exam Date:</b> 29/5/2015</p> <p><b>Description about the exam:</b> This exam can be given by any graduates,post graduates and engineers.</p> <p><b>Examination Fees:</b> Rs.500</p>	
<input type="button" value="Submit"/> <input type="button" value="Reset"/> <input type="button" value="Cancel"/>	

Purpose	:	Update Selected Exam Page
Functionality	:	<ul style="list-style-type: none"> <li>• The exam selected can be updated using this page. All the details entered previously for the selected exam gets populated in the page.</li> </ul>

Role	Details / Business Rule / Constraints			
Administrator	These users can invoke this functionality.			

Control Label	Control Type/ Length/ Multi select	Description/ Business Rule	Default Value	Validation
Back	Button	On click redirects to the exam selection page	-	-
LogOut	Button	On click logs out the admin from the portal	-	-
Exam Name	Text	Selected Exam name from the previous page gets reflected here	-	It should display the name of the exam to be updated. It is a non-editable field.
Exam Code	Text	Corresponding exam code is shown here	-	It should display the exam code of the exam to be updated. It is a non-editable field.
Bank Name	Text Box	Bank name can be changed or edited here	-	Should accept valid bank name.
Commencement of Online Registration	Calendar Control	Date can be changed or edited for the commencement of the online registration	-	Calendar should get populated while clicking on the control. It should be a future date.
Commencement of Online Application	Calendar Control	Date can be changed or edited for the commencement of the online application	-	Calendar should get populated while clicking on the control. Should be

				30 days from the commencement date.
Exam Date	Calendar Control	Date can be changed or edited for the exam	-	Calendar should get populated while clicking on the dropdown box. Should be more than 60 days from the date of commencement date.
Description about exam the exam	Text Box	Description about the exam can be changed or edited here	-	It can accept characters and numeric values.
Examination Fees	Text Box	The fee amount can be changed or edited here	-	It can accept only numeric values.
Submit	Button	On click it submits all the details	-	-
Reset	Button	On click it resets the details	-	-
Cancel	Button	On click cancels the operation and closes the web page	-	-

## Delete an exam Page

Banking Exam Portal	
<input style="border: 1px solid #ccc; padding: 2px; margin-right: 10px;" type="button" value="Back"/> <span style="font-size: 1.2em; font-weight: bold;">Delete an Exam</span> <input style="border: 1px solid #ccc; padding: 2px; margin-left: 10px;" type="button" value="LogOut"/>	
<b>Exam Name</b> <input style="border: 1px solid #ccc; padding: 2px; margin-right: 10px;" type="button" value="Select Exam Name"/>	
<input style="border: 1px solid #ccc; padding: 2px; margin-right: 10px;" type="button" value="Delete"/> <input style="border: 1px solid #ccc; padding: 2px;" type="button" value="Cancel"/>	

Purpose	:	Select Exam to be deleted
Functionality	:	<ul style="list-style-type: none"> <li>• The exam to be deleted can be selected in this page.</li> </ul>

Role	Details / Business Rule / Constraints
Administrator	These users can invoke this functionality.

Control Label	Control Type/ Length/ Multi select	Description/ Business Rule	Default Value	Validation
Back	Button	On click redirects to the home page of the admin	-	-
LogOut	Button	On click logs out the admin from the portal	-	-
Exam Name	Drop Down Box	Exam to be deleted can be selected from the list	-	All the exams available should be present in the list.
Delete	Button	On click it deletes the selected exam and the details of that exam from the database.	-	-
Cancel	Button	On click cancels the operation and closes the web page	-	-

## New Registration Page

Banking Exam Portal				
<input type="button" value="Back"/> <input type="button" value="New Registration"/> <input type="button" value="LogOut"/>				
Exam Applied For <input type="button" value="Select Exam Name"/>				
Exam Code :				
<b>Personal Details</b>				
First Name <input type="text"/>	Middle Name <input type="text"/>	Last Name <input type="text"/>		
Mobile Number <input type="text"/>	Confirm Mobile No <input type="text"/>			
Email Address <input type="text"/>				
Date of Birth <input type="button" value="Select Date"/>				
Gender <input checked="" type="radio"/> Male <input type="radio"/> Female				
Marital Status <input type="radio"/> Married <input checked="" type="radio"/> UnMarried				
<b>Educational Qualification Details</b>				
Graduation/Equivalent Passes <input type="button" value="Select Graduation"/> Degree/Stream Subject <input type="text"/> Year of Passing <input type="button" value="select Year"/> %of Marks <input type="text"/>				
<b>Photograph and Signature</b>				
Insert Signature <input type="file"/>	<input type="button" value="Browse..."/>	(Signature should be 10Kb to 20Kb)		
Insert Photograph <input type="file"/>	<input type="button" value="Browse..."/>	(Applicant's photo should be upto 50Kb)		
<input type="button" value="Save &amp; Submit"/> <input type="button" value="Reset"/> <input type="button" value="Cancel"/>				

Purpose	:	Registration page
Functionality	:	<ul style="list-style-type: none"> <li>A new applicant can register into the portal using this page</li> </ul>

Role	Details / Business Rule / Constraints
Applicant/User	These users can invoke this functionality.

Control Label	Control Type/ Length/ Multi select	Description/ Business Rule	Default Value	Validation
Back	Button	On click takes to the main home page of the portal	-	-
LogOut	Button	On click logs out the applicant from the portal	-	-
Exam Applied For	Drop Down Box	Exam name can be selected from the drop down list	-	Valid exam names should get displayed

				in the drop down list
Exam Code	Text Box	Corresponding exam code is shown here	-	The exam code of the selected exam should be displayed. It is a non-editable field.
First Name	Text Box	First name of the applicant can be given here	-	Should accept only alphabets.
Middle Name	Text Box	Middle name of the applicant can be given here	-	Should accept only alphabets.
Last Name	Text Box	Last name of the applicant can be given here	-	Should accept only alphabets.
Email Address	Text Box	Email address of the applicant can be given here	-	It can accept characters, numeric values, _ and should end with a domain name from the values of gmail.com, yahoo.com and Hotmail.com.
Date of Birth	Calendar Control	Date can be selected for the date of birth	-	Calendar should get populated while clicking on the control.
Gender	Radio button	Applicant can select any one from the two options given	-	Two radio buttons to be provided
Marital Status	Radio button	Applicant can select any one from the two given options	-	Two radio buttons to be

				provided
Graduation /Equivalent Passes	Drop Down Box	Appropriate qualification can be selected from the drop down list	-	All relevant qualifications should be mentioned in the drop down list
Degree/Stream/Subject	Text Box	Appropriate subject of the degree can be mentioned here	-	Should accept only characters
% of marks	Text Box	Marks of the applicant can be mentioned here	-	Should accept only numeric values
Insert Signature	File Control	The image file for signature of the applicant can be selected and uploaded here	-	It should allow to select the image from the place where it is stored in the system. The image file should be between 10Kb and 20Kb.
Insert Photo	File Control	The image file for photo of the applicant can be selected and uploaded here	-	It should allow to select the image from the place where it is stored in the system. The image file should not be more than 50Kb.
Save & Submit	Button	On click it submits all the details	-	-
Reset	Button	On click it resets the details	-	-
Cancel	Button	On click cancels the operation and closes the web page	-	-

## Applicant Home Page

<b>Banking Exam Portal</b>		
Back	<b>Welcome Applicant!</b>	LogOut
<a href="#">Make Payment</a> <a href="#">Print Receipt</a> <a href="#">View Application</a>		

Purpose	:	Applicant's home page
Functionality	:	<ul style="list-style-type: none"> <li>The applicant's functionality of making payment, printing the receipt and viewing the application can be accessible from this page</li> </ul>

Role	Details / Business Rule / Constraints
Applicant/User	These users can invoke this functionality.

Control Label	Control Type/Length/ Multi select	Description/ Business Rule	Default Value	Validation
Back	Button	On click redirects to the main home page of the portal	-	-
LogOut	Button	On click logs out the applicant from the portal	-	-
Make Payment	Hyperlink	Applicant can make the payment for an exam using this link	-	It should navigate the new user to the <b>Make Payment</b> web page
Print Receipt	Hyperlink	Applicant can print the receipt using this link	-	It should navigate the new user to the <b>Print Receipt</b> web page
View Application	Hyperlink	Applicant can view the application using this link	-	It should navigate the new user to the <b>View Application</b> web page

## Payment Page

**Payment Option**

Payment Mode :  Debit Card

Select IBank :

**Payment Information**

**Card Details**

Card Number

Valid Thru / Expiry Date  MM  YYYY

Cardholder's Name

PIN

Enter the characters visible in the box below

**2 E F L 2**

Purpose	:	To make the payment for the exam applied
Functionality	:	<ul style="list-style-type: none"> <li>The payment for the exam applied can be done through this page</li> </ul>

Role	Details / Business Rule / Constraints
Applicant/User	These users can invoke this functionality.

Control Label	Control Type/Length/Multi select	Description/Business Rule	Default Value	Validation
Payment Mode	Radio Button	The payment mode of 'Debit Card' can be selected	-	Should have only 1 value of Debit Card
Select Bank	Drop Down Box	Bank name can be selected from the list of names	-	Should contain valid bank names in the list
Card Number	Text Box	Debit card number of the applicant can be mentioned here	-	It should accept only 14 digit numeric values.
Valid Thru/Expiry	Drop down Box	Expiry month of the	-	MM should

date (MM)		Debit Card can be selected		contain values from 01 to 12.
Valid Thru/Expiry date (YYYY)	Drop down Box	Expiry year of the Debit Card can be selected		YYYY should contain values from 2000 to 2020
Cardholder's Name	Text Box	Name of the cardholder can be mentioned	-	It should accept only characters
PIN	Password Box	PIN number can be mentioned here	-	It can accept only 4 digit numeric values
Examination Fees	Text Box	The fee amount can be entered here	-	It can accept only numeric values
Enter the characters visible in the box below	Text Box	The characters visible in the box can be entered here	-	It can accept both numeric and alphabets
Make Payment	Button	On click it makes the payment for the exam	-	-
Back	Button	On click redirects to the applicant's home page of the portal		

## Receipt Page

Banking Exam Portal		
<a href="#">Back</a>	<b>Receipt</b>	<a href="#">LogOut</a>
<p>Registration No: 1356689            Applicant Name: Jamie Joy            Date of Birth: 06/05/1984            Exam Applied For: AXISPOOL            Date of Transaction: 17/06/2015            Transaction Amount: Rs. 500            Transaction ID: 8734538720475032            Mobile No: 9820238592</p> <p><a href="#">Print Receipt</a>    <a href="#">Cancel</a></p>		

Purpose	:	To view the receipt for the payment made and a print can be taken
Functionality	:	<ul style="list-style-type: none"> <li>The receipt for the payment made can be viewed through this page along with print receipt functionality</li> </ul>

Role	Details / Business Rule / Constraints
------	---------------------------------------

Applicant/User	These users can invoke this functionality.
----------------	--

Control Label	Control Type/ Length/ Multi select	Description/ Business Rule	Default Value	Validation
Back	Button	On click redirects to the applicant's home page	-	-
LogOut	Button	On click logs out the applicant from the portal	-	-
Print Receipt	Button	On click prints the receipt visible on the page	-	-

### **View Application Page**

Purpose	:	To view the application details
Functionality	:	<ul style="list-style-type: none"> <li>• The applicant can view the application details through this page</li> </ul>

Role	Details / Business Rule / Constraints
Applicant/User	These users can invoke this functionality.

### **Sample Application Details Page**

Registration Details			
Exam Applied For	AXISPOO1		
<b>Personal Details</b>			
Full Name	Jamie Joy		
Mobile No	9820238592		
Email Address	Jamie.joy@gmail.com		
Date of Birth	06-05-1984		
Gender	Female		
Marital Status	Married		
<b>Education Qualification Details</b>			
Graduation/Equivalent Passed	Degree/Stream Subject	Year of Passing	% of Marks
Bachelor of Engineering	Information Technology	2006	78
<b>Application Fees Details</b>			
Payment Mode	Online		
Payment Status	Paid		
Transaction Date	17/06/2015		
Amount	Rs.500		
<b>Declaration</b>			
I hereby declare that all the statements made in this application are true, complete and correct to the best of my knowledge and belief. I understand that in the event of any information being			

found untrue or incorrect at any stage or I am not satisfying any of the eligibility criteria stipulated, my candidature is liable to be cancelled. In case of creating influence, candidature of the candidate shall be cancelled.

**Date: 17-06-2015****Signature of the applicant**

Purpose	:	The entire application can be viewed
Functionality	:	<ul style="list-style-type: none"><li>• The applicant can view the application and the details entered during registration and payment through this page.</li></ul>

## Lab 2. Identify ambiguities in the requirements

<b>Goals</b>	<ul style="list-style-type: none"> <li>In this lab you'll perform a requirements review for QWERTYsoft by reviewing the QWERTYsoft Requirements Document.</li> <li>Identify problems with specific phrases and sections.</li> <li>Try to fix those problems by rewording the phrase, changing the sentence, adding new sentences, or even whole new paragraphs.</li> </ul>
<b>Time</b>	120 Minutes

### 1. Scope QWERTYsoft - Requirements Document

This document specifies the requirements for a family of Internet kiosks called QWERTYsoft. These kiosks shall provide cash, credit card, and debit card customers with simple, fast, reliable Internet access in public places for reasonable prices per minute of usage.

#### 1.1 Terms, Acronyms, and Abbreviations

For the purposes of this project, the following abbreviations apply:

AS	Application Server
Cable	Cable high-speed Internet connection at least 128Kbps
CC	Credit card (for payment): American Express, Visa, or MasterCard
CS	Communication Server
DBMS	Database Management System (server)
DC	Debit card (for payment): PLUS or Cirrus networks.
DSL	Digital Subscriber Line high-speed Internet connection (either asymmetric or symmetric) at least 128Kbps
IE	The Internet Explorer Internet browser
Kbps	Kilobits per second
Kiosk	The free-standing QWERTYsoft Internet access point
Linux	Red Hat Linux Release 8.0 operating system
Opera	The Opera freeware Internet browser
PIN	Personal Identification Number (for debit card)
PSTN	Public Switched Telephone Network Internet connection (ordinary dial-up connectivity) at least 50 KBPS
URL	Universal resource locator

WS	Web Server
WXP	Windows XP Professional operating system

## 2. Required release date

The first set of 1,000 QWERTYsoft kiosks shall be live, accepting payment and accessing the Internet, as of the financial third quarter.

## 3. Description of requirements

### 3.1 General technical requirements

QWERTYsoft shall give customers in airports, malls, theaters, and other public places access to the Internet. QWERTYsoft shall provide call center agents with access to information about current and past kiosk sessions as well as the ability to control current sessions.

## 4. Welcome

Between sessions, each QWERTYsoft kiosk shall display an inviting welcome message (see screen prototype K.1).

## 5. Payment

Once a user navigates past the Welcome screen, the kiosk shall give the user the option to purchase a block of browser time in the Payment screen (see screen prototype K.2). The kiosk shall sell blocks of time in five (5) minute increments, up to one (1) hour.

The system accepts the following forms of payment:

- a) Cash (bills only) (see screen prototype K.3)
- b) Credit card (American Express, Visa, or MasterCard only) (see screen prototype K.4 and K.7)
- c) Debit card (PLUS or Cirrus networks only) (see screen prototype K.5 and K.7)

Once the current block of time is within sixty (60) seconds of expiration, the kiosk shall pop up a message asking if the user wants to purchase more time (see screen prototype K.9).

## 6. Internet Browser

At the Welcome screen, each QWERTYsoft kiosk shall provide the user with a choice of the latest version of Netscape, Opera, or Internet Explorer (available on Windows kiosks only).

## 7. Performance

On kiosks operating with a PSTN connection, users shall have greater than 50Kbps connection speed.

On kiosks operating with DSL or cable connections, users shall have greater than 128Kbps connection speed.

## 8. Localization

Each QWERTYsoft kiosk shall be configured to operate in the primary local language for its installed locale.

In locales where multiple languages are commonly used, the Welcome screen shall present the user with the option to select the language for the session.

Each QWERTYsoft kiosk browser shall be configured to support all languages supported by the operating system and browser.

## 9. Content Control

Because QWERTYsoft users will access the Internet in public places, QWERTYsoft shall implement site blocking that prevents the display of pornographic, objectionable, lewd, obscene, or violent material.

QWERTYsoft shall protect each kiosk against sending or receiving a virus, worm, or other malicious code.

## 10. Session Termination

Users may terminate sessions in one of two ways:

- a) Logging out (no refund is given for unused time)
- b) Allowing time to expire.

## 11. Confidentiality

To protect user confidentiality — e.g., URLs visited — once a session terminates, each kiosk shall clear all cookies and other downloaded files, clear the URL history, exit the browser, and restart the browser at the Welcome screen.

## 12. Administration

### a) Software Updates

Under ordinary circumstances, software updates will take place automatically. At 2:00 A.M. local time, each kiosk shall connect to the server farm and ask for updates. Those updates include:

- Operating system or browser patches
- New network, modem, or graphics drivers
- New logos
- Updated per-minute payment rate tables
- Virus, worm, malicious code, or other firewall definitions
- Blocked websites.
- If there are no updates available, the kiosk shall disconnect.

If the update application on the application server tells the kiosk that it is overloaded, the kiosk shall disconnect, then retry at a later time. The delay for retry is a random period between ten (10) and sixty (60) minutes.  
Call center agents may also push software updates to kiosks.

### **13. View Kiosks**

Call center agents shall be able to browse a list of kiosks. For each kiosk, call center agents shall be able to see:

- Current operating system version
- Current browser version
- Total uptime since installation
- Total uptime since last software update
- Number of crashes, reboots, or other hard failures since last software update.

Kiosks shall connect to the server farm once per hour to report status. If a kiosk is not connected to the server farm, the call center agent may force a connection to check status.

If a kiosk is down, that kiosk shall show up at the top of the list of kiosks, highlighted in red.

### **14. View Users**

For those kiosks that have active users, call center agents shall have access to the following information:

- Current and past URLs.
- Credit or debit card number (if applicable)
- Name (if available from credit card validation)
- Amount paid for this session
- Blocks of time purchased
- Previous session (if available from credit card number and name)
- Paid time remaining

### **15. Modify User**

Call center agents shall be able to modify a user's session by adding blocks of time. Supervisory override is required for an agent to add more than sixty (60) minutes of time per day.

### **16. Terminate User**

If a call center agent believes that a user is engaged in illegal, inappropriate, or fraudulent use of a session, the agent may terminate that session.

The user shall receive a refund for any unused time at the point of termination. The user shall receive a message that the session was terminated for inappropriate activity. The message shall specify the amount of the refund.



## REQUIREMENTS VALIDATION AND FUNCTIONAL DECOMPOSITION LAB BOOK

**Note – Use the following Template to complete this lab.**

Continued