Testing Concepts

Lesson 6: **Requirement Engineering**

Capgemini

Lesson Objectives

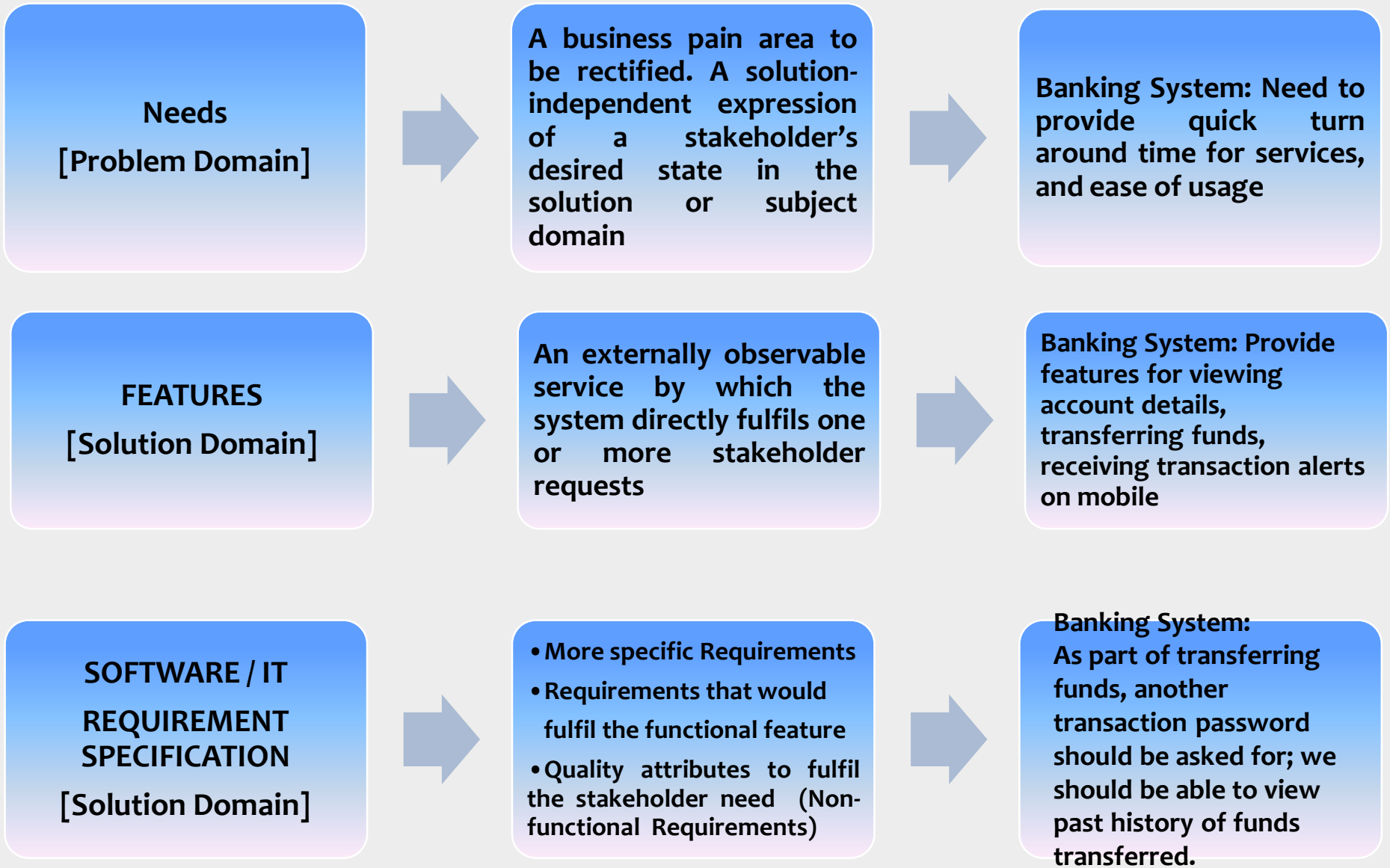
To understand the following topics:

- Evolution of Requirements
- Who provides the Requirements?
- Challenges in Requirement Gathering
- Why do we need good requirements?
  - Characteristics & Impact of bad Requirements
- Requirement engineering
- Functional Vs Non-Functional Requirements
- Non Functional Requirements: FURPS +

Lesson Objectives

- Stable and Volatile Requirements
- Baselining Requirements
- Requirements Traceability
  - Requirement Traceability Matrix
  - Maintaining Requirement Traceability
  - Requirement Traceability Matrix – Example
- Requirements Change
  - Change Management Process
  - Requirement Creep

6.1 Evolution of Requirements



6.2 Who provides the Requirements? - Stakeholders

"Stakeholders" are the individuals who affect or are affected by the proposed software product.

Following are the different stakeholders :

- Customers – These people purchase, and/or pay for the software product in order to meet their business objectives
- End Users – use the product directly or indirectly by receiving reports, outputs, or other information generated by the product
- Development Team – They include individuals/teams from the development organization :
  - Business Analyst - elicit the requirements from the customers, users, and other stakeholders; analyze requirements, write requirements specification, and communicate the requirements to development team and other stakeholders.
  - Designers –translate the requirements into the software's architectural and detailed designs specifying how the software will be implemented.
  - Developers – implement the designs by creating the software product
  - Testers – They use the requirements for designing the test cases that they use to execute and test the software under specific, known conditions to detect the defects and provide confidence that the software performs as specified.

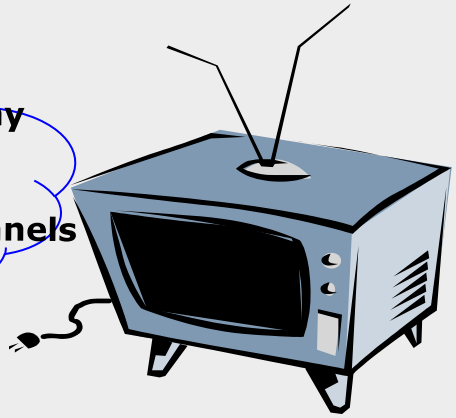
Different stakeholders….So different requirements

Plasma TV -Sharp picture
AV in and out
Surround sound
Low Price

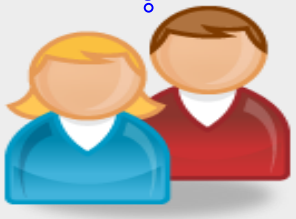
Easy to fix repairs

Two Remotes!
PIP

Box should match my drawing room color
Lock children's channels



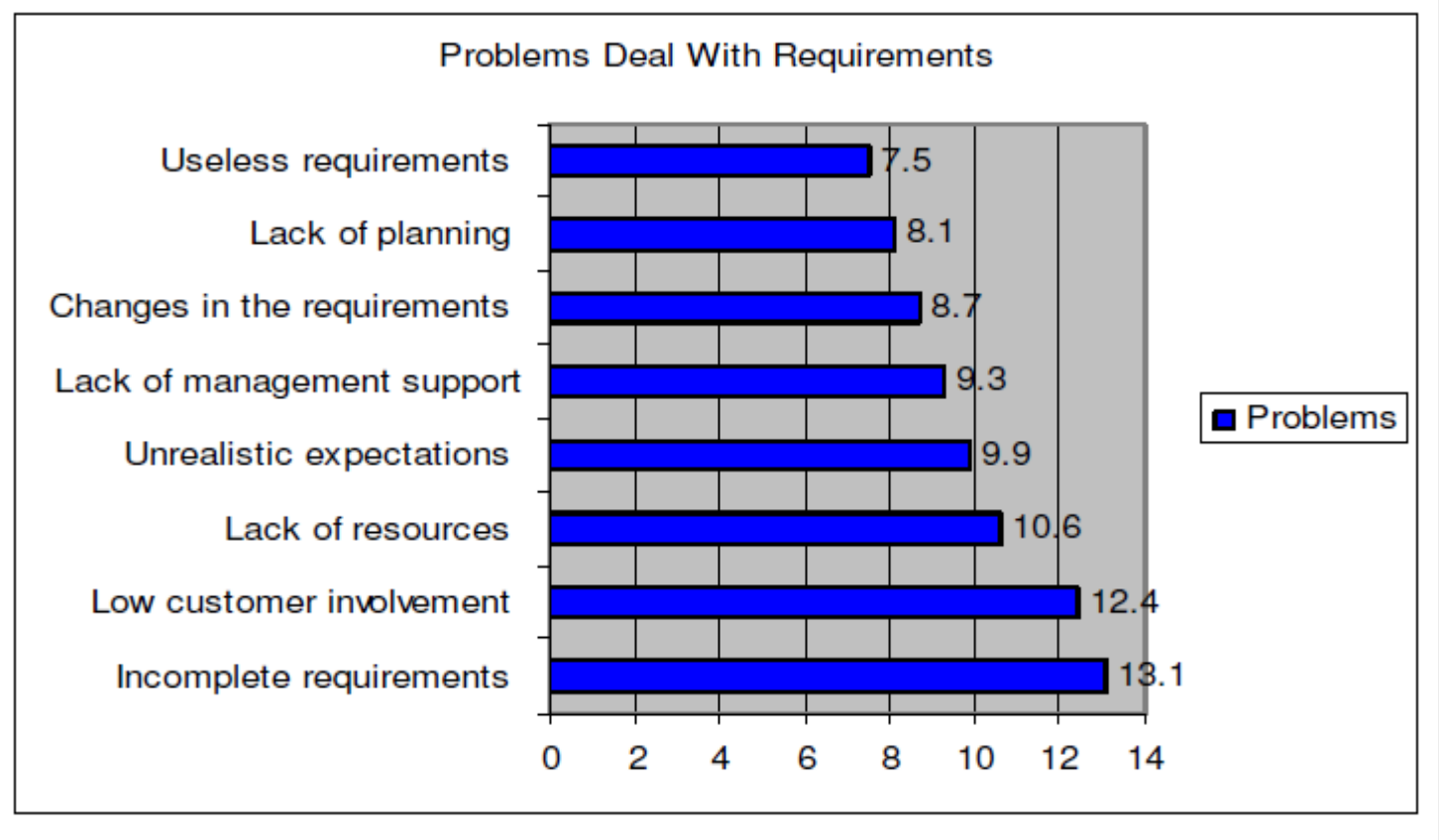
Hope it'll be easy to operate



6.3 Challenges in Requirement Gathering

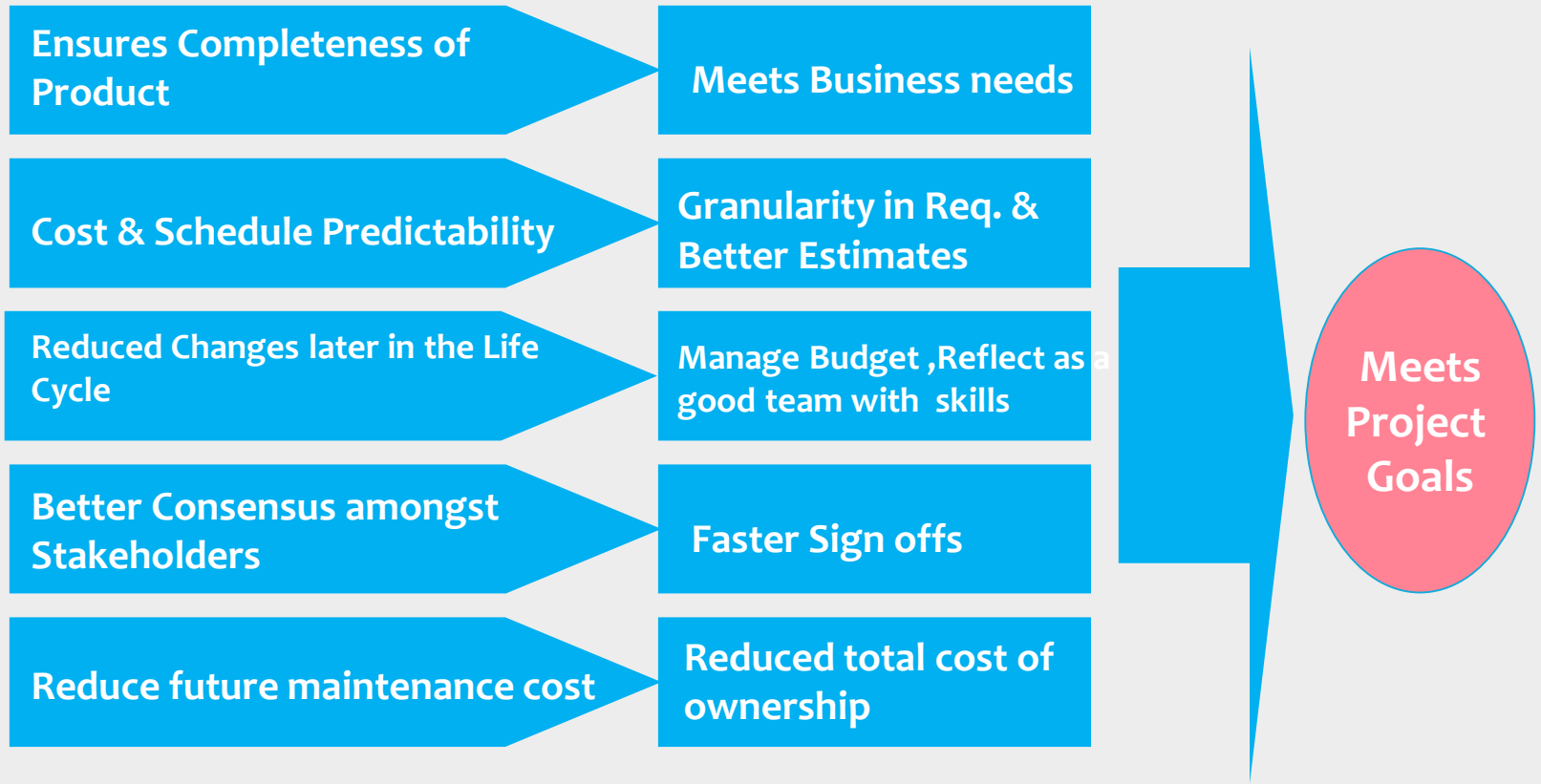
Requirement Problems are the single No.1 reason for projects to fail.

According to the Standish Group's 1995 CHAOS survey, the top two "project impaired" factors were **incomplete requirements** and **lack of user involvement.**



6.4 Why do we need good requirements?

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



6.4.1 Characteristics & Impact of bad Requirements

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

Impact of bad Requirements :

- Function failure
- Extensively over budget
- Extensively past schedule
- Extensively reduced scope
- Poor quality applications
- Not considerably used when delivered
- Sometimes getting cancelled

6.5 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

6.6 Functional & Non-functional Requirements

Functional Requirements	Non-functional Requirements
It specifies the input and output behaviour of a systems. It defines how software behaves to meet user needs. Example : Functional requirements of a health insurance company include : • Determining Claimant Eligibility • Paying Claims • Calculating Premium	It represents quality attributes of the system : Availability, Maintainability, Performance, Portability, Reliability, Robustness, Security, Scalability etc. 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

Functional Vs Non-Functional Requirements

6.7 Non Functional Requirements: FURPS +

Functionality
Usability
Reliability
Performance
Supportability
+ other such quality attributes

6.8 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 been 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

6.9 Baselining Requirements

The requirements are baselined at the end of the Requirements Development phase & ideally signed-off by the customer

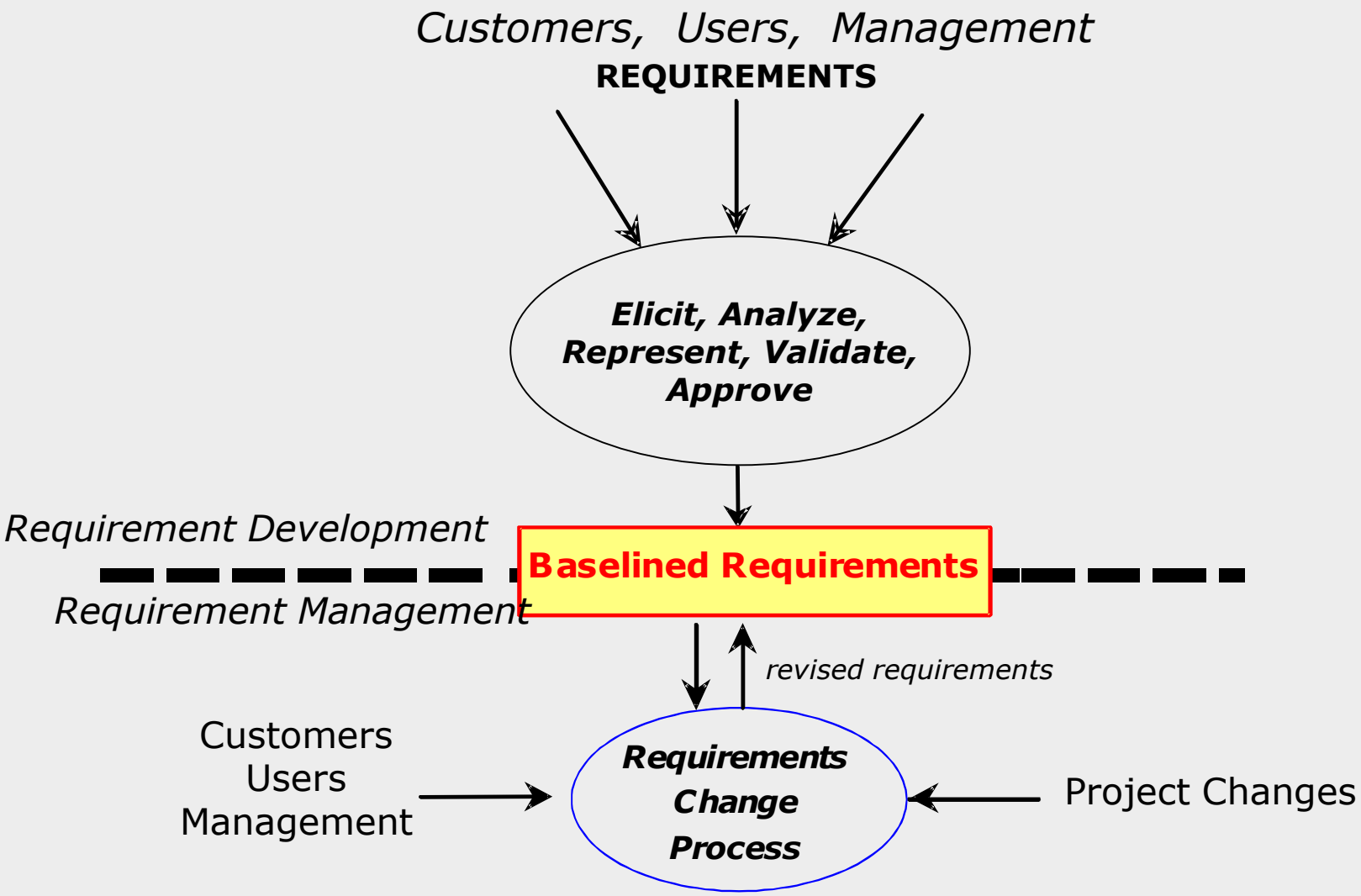
A requirements baseline is:

- 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

Baselining Requirements (Cont..)



6.10 Requirements Traceability

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***

6.11.1 Requirement Traceability Matrix

RTM 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

Suggested Tools

- Rational Requisite Pro
- Excel Sheet (Traceability template)

6.11.2 Requirement Traceability Matrix – Example

Example : XYZ Banking Application

This is a site that aims at computerizing the business process of XYZ bank. The following are the business targets that are to be achieved :

Step 1: Business Requirement Document (BRD) :

BR ID	Module Name	Roles Involved	Description
BR_1	Login	Customer Manager	A customer can login using login module. A Manager can login using login module.
BR_2	Enquiry	Customer Manager	A customer can view balance of his accounts only. A manager can view balance of all customers who come under his supervision.
BR_3	Fund Transfer	Customer Manager	A customer can transfer funds from his 'own' account to any destination account. A manager can transfer funds from any account under his supervision.
BR_4	Loan Process	Customer Manager	A customer can access loan information and apply for loan. A manager can grant, reject, suggest changes to the loan request under his supervision.

Requirement Traceability Matrix – Example

Below is our Technical Requirement Document (TRD) based on the interpretation of the Business Requirements Document (BRD)

Step 2: Technical Requirement Document (TRD) for **BR_1 (Login)**

TR ID	Module Name	Description
TR_01	Login Module	User ID and Pwd must not be blank
TR_02	Login Module	If User ID and password are valid. Login.
TR_03	Login Module	The Pwd field should not allow copy paste from any source.

Requirement Traceability Matrix – Example

Step 3: On the basis of Business Requirement Document (BRD) and Technical Requirement Document (TRD), testers start writing test scenarios.

Test Scenarios :

Test Scenario ID	Test Scenarios
TS_1	Verify with valid User Credentials
TS_2	Verify with invalid User Credentials
TS_3	Verify the length provided for the user name field
TS_4	Verify the length provided for the Pwd field
TS_5	Verify copying the Pwd from external file and pasting it into Pwd field.

Requirement Traceability Matrix – Example

Step 4: For each test scenario, write at least 1 or more test cases.

Test Cases :

Test Case ID	Test Condition	Test Steps	Test Data	Expected Result
TC_1	To validate that user is able to login successfully with valid User ID & valid Pwd.	1.Go to Login Page 2.Enter User ID 3.Enter Password 4. Click Login	user1 1234B	Login Successful
TC_2	To validate that user is unable to login with invalid User ID & valid Pwd.	1.Go to Login Page 2.Enter User ID 3.Enter Password 4. Click Login	sampleuser 1234B	Login Failure
TC_3	To validate that user is unable to login with valid User ID & invalid Pwd.	1.Go to Login Page 2.Enter User ID 3.Enter Password 4. Click Login	user1 samplePwd	Login Failure

Requirement Traceability Matrix – Example

Step 5: You can now start creating RTM.

Identify the Test scenarios, Technical Requirements and Business Requirements that the test cases are verifying.

BR ID	TR ID	TS ID	TC ID
BR_1	TR_01	TS_3	
		TS_4	
	TR_02	TS_1	TC_1
		TS_2	TC_2
		TS_2	TC_3
	TR_03	TS_5	
BR_2			
BR_3			
BR_4			

At this stage, the RTM can be used to find gaps. For example, in the above RTM, you see that there are no test cases written for TS_3, TS_4, TS_5.

it will indicate the places where the test team needs to work some more to ensure 100% coverage.

Requirement Traceability Matrix – Example

Step 6: Expand the RTM to include test case execution status and defects.

BR ID	TR ID	TS ID	TC ID	Status	Defect ID
BR_1	TR_01	TS_3	TC_5	Pass	
		TS_3	TC_6	Fail	D_05
		TS_4	TC_7	Pass	
	TR_02	TS_1	TC_1	Pass	
		TS_2	TC_2	Fail	D_11
		TS_2	TC_3	Pass	
	TR_03	TS_5	TC_10	Fail	D_02

6.11 Requirements Change

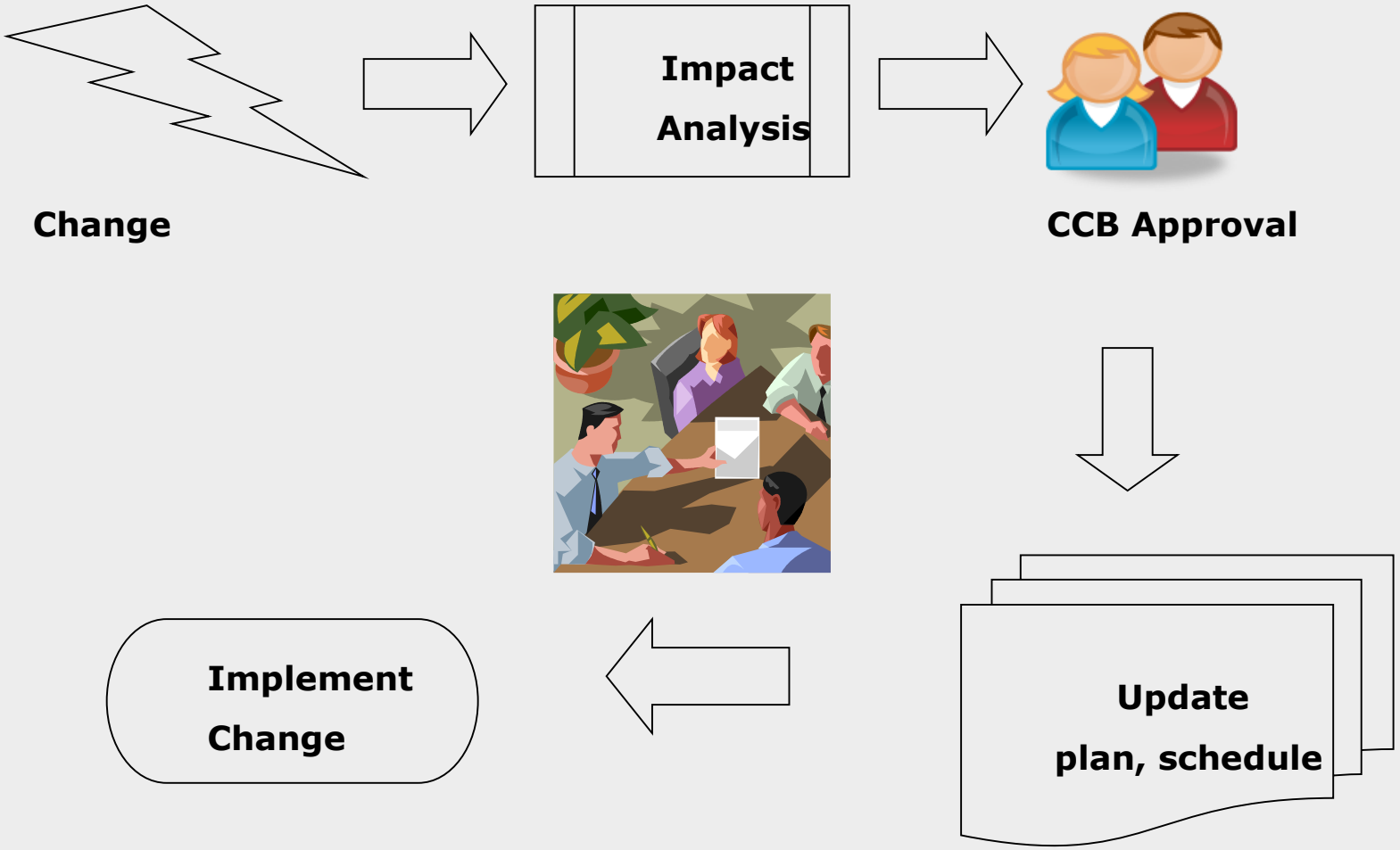
Users **can** propose **requirements change** at any stage of **SDLC.**

Requirements change because :

- Understanding of the problem improved : Customer's expectations change once they see the product taking shape
- Initial elicitation activities are imperfect : We failed to ask the right people the right questions at the right time.
- The priority of requirements from different viewpoints change during the development process
- Business needs evolve
- The users changed their perceptions : Customers may specify requirements from a business perspective that conflict with end-user requirements
- The external environment changed : The business and technical environment of the system changes during its development
- We either failed to create or follow a process to help manage change.

6.11.1 Change Management Process

Establish a formal process for managing changes to the system requirements



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

6.11.2 Requirement Creep

"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."

Why does Requirement Creep occur ?

- Poor requirement analysis
- Not involving customers early though
- Insufficient detailing on the complexity of the project
- Lack of change control
- Gold Plating
- Unwillingness to say no to a client

Requirement Creep (Cont..)

- Even when there is a clearly defined project scope, one must be aware that Requirement creep can still occur during project development.
- Requirement creep tends to additional requirements needed to achieve the new objectives. This 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.

Measures to control Requirement Creep

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

Review Question

Question 1: Which of the non functional requirements can ensure that the application should be available for German & Japanese users?

Question 2: In which of the requirement gathering patterns, requirements are specified in detail and passes thru multiple reviews and sign-offs?

Question 3: Volatile Requirements are likely to change during the system development process or after the system has been become operational. (T/F)

Question 4: The requirements are baselined at the beginning of the Requirements Development phase & ideally signed-off by the development team. (T/F)

Question 5: Which type of requirement traceability is used to validate whether the project is evolving in the desired direction and for the right product?