
Req. Validation & Functional Decomposition for V&V Automation Testing
Lesson 1: Introduction to Requirements Engineering
Capgemini

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

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)

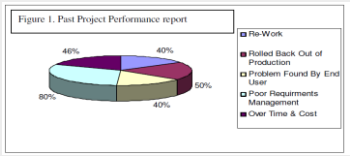
Why do Projects fail? – Current Survey

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]

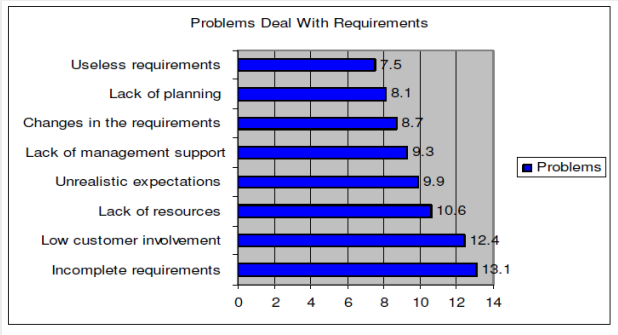
Standish Report

- 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



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:

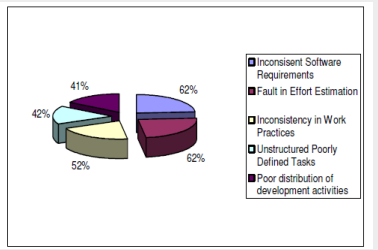


Why do Projects fail? – Current Survey

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



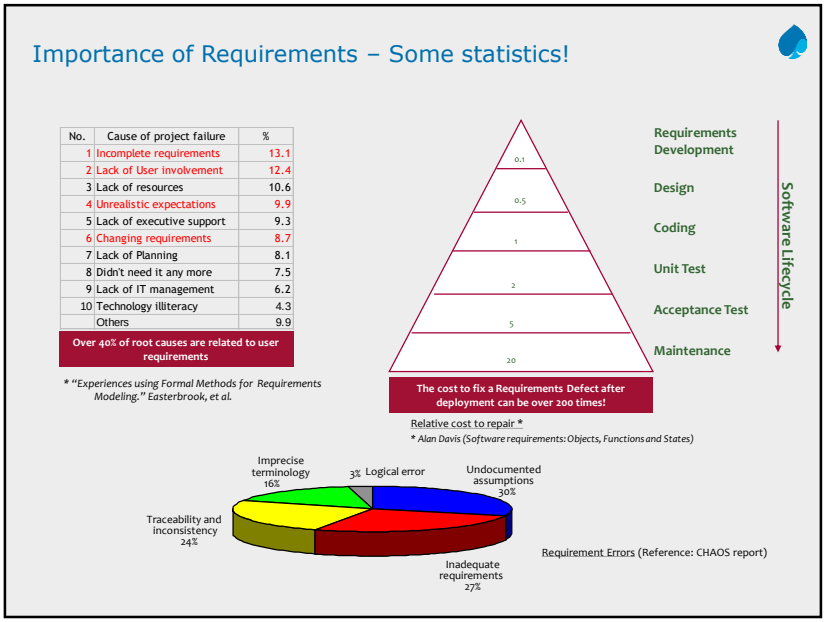
Requirements Contains Defects

The percentage of defects originating during requirement engineering are estimated as :

- 50% - Karl Wiegers 2001
- 42% - A. Winngrove
- 60%-64% - requirements and design EBG Consulting

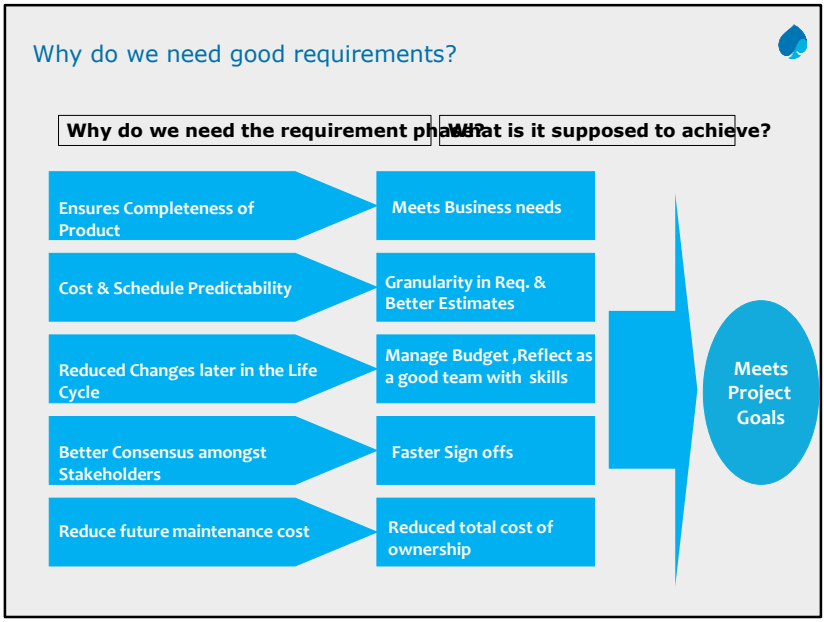
Characteristics of defective requirements:

- Lack of Cohesiveness
- Lack of Completeness
- Lack of Correctness
- Lack of Consistency
- Lack of Project Relevance
- Lack of Testability, Usability, Validatability
- Ambiguous



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?
Ensures Completeness of Product
Meets Business needs
Cost & Schedule Predictability
Granularity in Req. & Better Estimates
Reduced Changes later in the Life Cycle
Manage Budget ,Reflect as a good team with skills
Better Consensus amongst Stakeholders
Faster Sign offs
Reduce future maintenance cost
Reduced total cost of ownership
Meets Project Goals

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]

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) ?

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

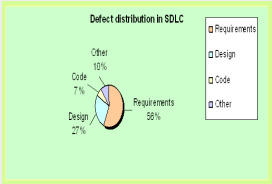
> ***"If you don't get the requirements right, it doesn't matter how well you do anything else."***
>
> ***Karl Wiegers (2004)***

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



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)

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.

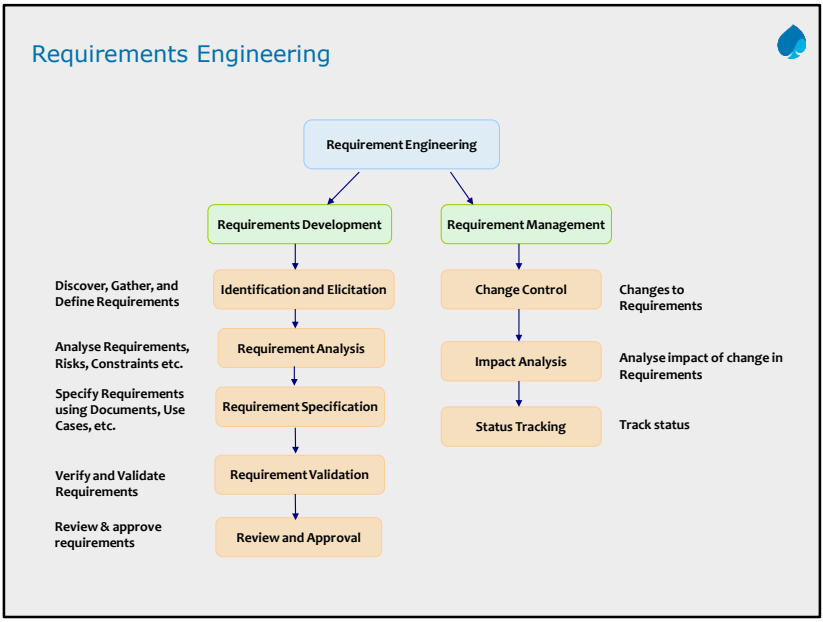
The gradual stabilization of such models in terms of the requirements leads to a satisfactory candidate specification, which then must be validated and verified. This gives stakeholders feedback on the interpretation of their requirements so they can correct misunderstandings. Requirements engineering 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



Requirement Engineering Process :

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

1. **Requirement Development** : Requirements development involves all of the activities that are part of eliciting, analyzing, specifying, and validating the requirements
    a. **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.
    b. **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.
    c. **Requirement Specification:** The stage in which the requirements are formally documented is called Requirement Specification.

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

Answers:
Question1:
Option1

Question2:
Option1

Question3:
True

Question4:
Requirement Development

Question5:
Requirement Validation

Review Question

Question1: 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

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.