

COURSE NAME

SOFTWARE
ENGINEERING

CSC 3114

(UNDERGRADUATE)

CHAPTER 4

SOFTWARE REQUIREMENTS ENGINEERING

M. MAHMUDUL HASAN

ASSISTANT PROFESSOR, CS, AIUB

Web: <http://cs.aiub.edu/profile/m.hasan>

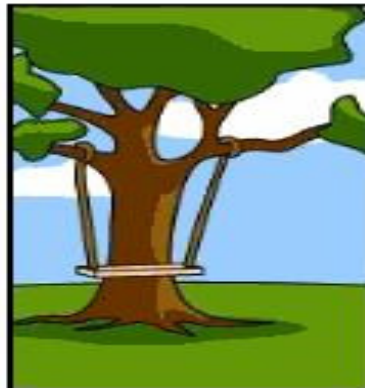
WordPress: <https://wordpress.com/view/mhasansuman.wordpress.com>



THE CURRENT COMMON PROBLEMS



How the customer explained it



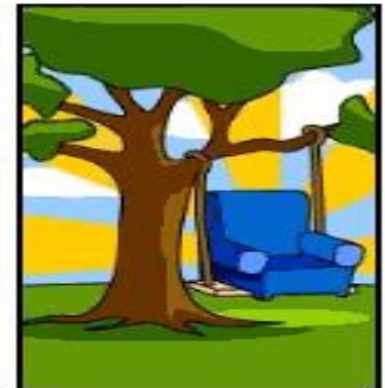
How the Project Leader understood it



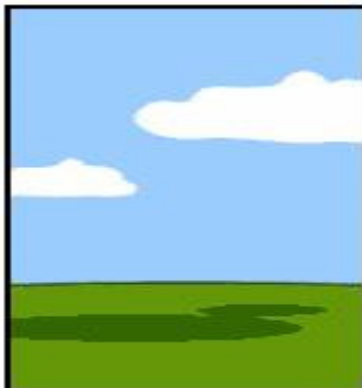
How the Analyst designed it



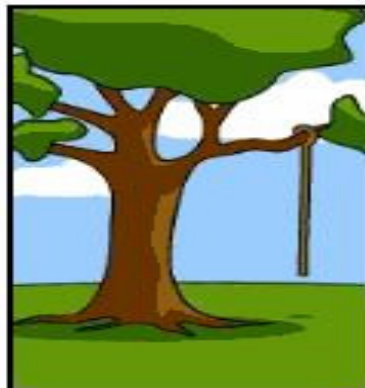
How the Programmer wrote it



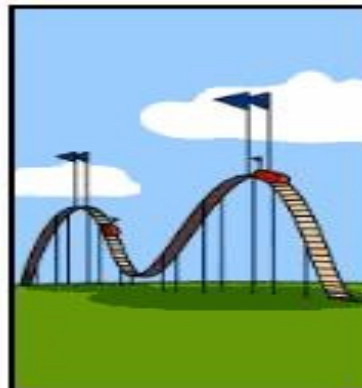
How the Business Consultant described it



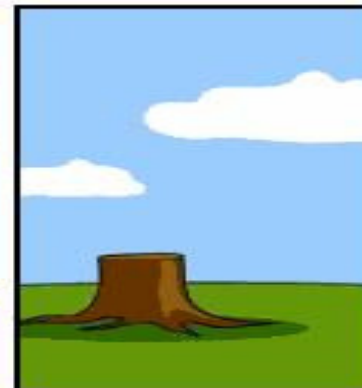
How the project was documented



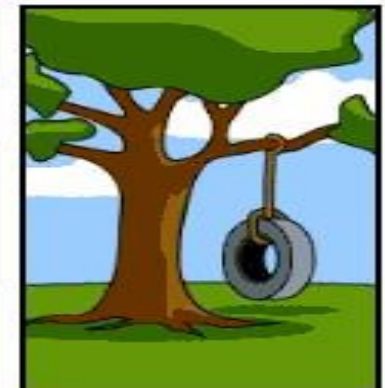
What operations installed



How the customer was billed



How it was supported



What the customer really needed

WHAT IS “REQUIREMENT”?

- A requirement is a property that a product must have to provide value to a stakeholder
- Requirements are a specification of **what** should be implemented. They are descriptions of a system property or attribute or **how** the system should behave. They may be a constraint on the development process of the system.
- Requirements encompass
 - the **user's view** of the **external system behaviour**
 - the **developer's view** of some **internal characteristics**
- Software requirements include a time dimension
 - They could be present tense, describing the current system's capabilities
 - Near-term (high priority) or hypothetical (low priority) future
 - They could even be past tense, referring to needs that were once specified and then discarded

BUSINESS REQUIREMENT

- Business requirements describe why the organization is implementing the system — the **business benefits** the organization hopes to achieve
- The focus is on the business objectives of the organization or the customer who requests the system (**business requirements collected from multiple sources might conflict**)
- *Suppose an airline wants to reduce airport counter staff costs by 25 percent. This goal might lead to the idea of building a kiosk that passengers can use to check in for their flights at the airport.*
- Self-service technologies (SSTs) have been applied to many areas of business
- Business requirements typically come from the **funding sponsor** for a project, the acquiring customer, the manager of the actual users, the **marketing department** (to promote a product or service), or a product visionary

USER REQUIREMENT

- ❑ User requirements describe **goals or tasks** the users must be able to perform with the product that will provide value to someone
 - Includes descriptions of product attributes or characteristics that are important to user satisfaction
 - Represent user requirements include use cases and **user stories**
 - User representatives will provide this information
 - Describes what the user will be able to do with the system
 - *An example of a use case is “Check in for a flight” using an airline’s website or a kiosk at the airport. Written as a user story, the same user requirement might read: “**As a passenger, I want to check in for a flight so I can board my airplane.**”*

FUNCTIONAL REQUIREMENT

- ❑ Functional requirements specify the behaviours the product will exhibit under specific conditions
 - They describe what the developers must implement to enable users to accomplish their tasks (user requirements), thereby satisfying the business requirements
 - Functional requirements often are written in the form of the traditional “**shall**” statements:
“The Passenger shall be able to print boarding passes for all flight segments for which he has checked in” or “If the Passenger’s profile does not indicate a seating preference, the reservation system shall assign a seat.”
 - A restriction that is imposed on the choices available to the developer for the design and construction of a product, is a **CONSTRAINT**
“The system shall be developed using open-source tools and shall run on Linux OS”

NON-FUNCTIONAL/QUALITY REQUIREMENTS

- Non-functional requirement are also known as quality attributes, product requirements that describes a service or performance characteristic of a product (“– ity, ilities.”)
- Describe the product’s characteristics in various dimensions that are important either to users or to developers and maintainers, such as performance, safety, availability, and portability (chances of conflicts within non-functional requirements are fairly high)
- Other classes of non-functional requirements describe external interfaces between the system and the outside world (usefulness, flexibility, reliability)
- These include connections to other software systems, hardware components, and users, as well as communication interfaces (interoperability)
- Design and implementation constraints impose restrictions on the options available to the developer during construction of the product (security)

PRODUCT VS. PROJECT REQUIREMENTS

- Requirements that describe properties of a software system to be built are called product requirements.
- Projects certainly do have other expectations and deliverables that are not a part of the software the team implements, but that are necessary to the successful completion of the project as a whole. These are project requirements but not product requirements. (maintain schedule deadline of task)
- An SRS houses the product requirements, but it should not include design or implementation details (other than known constraints), project plans, test plans, or similar information.
- Separate out such items so that requirements development activities can focus on understanding what the team intends to build.

REQUIREMENT ENGINEERING PROCESS

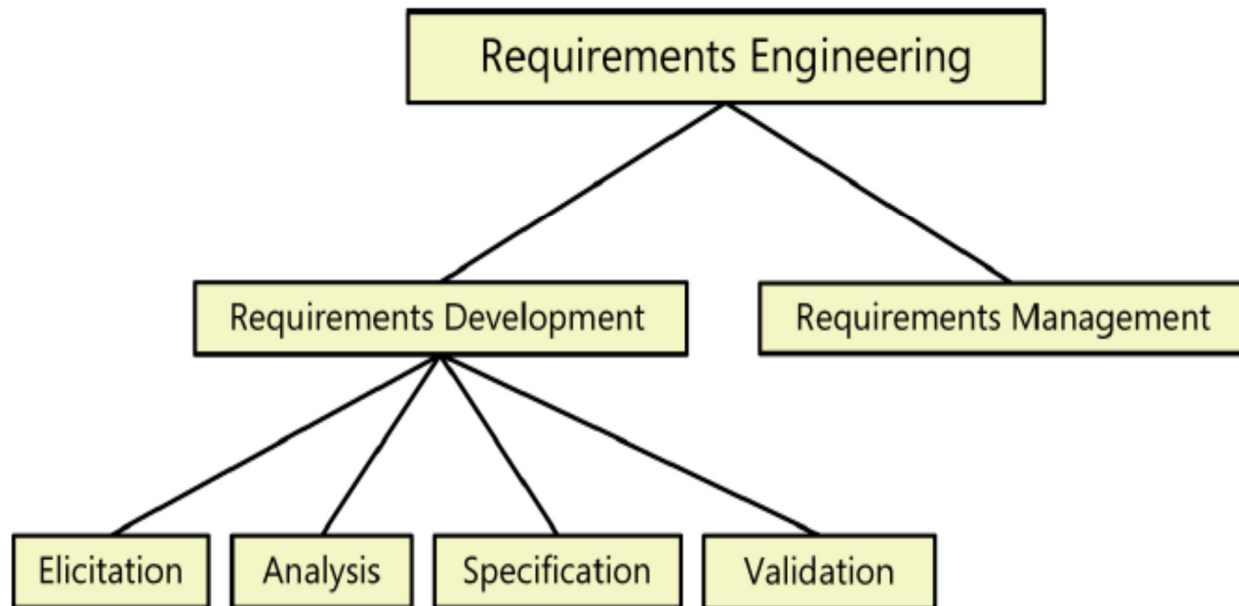


FIGURE 1-4 Subdisciplines of software requirements engineering.

□ Requirements Development

- Inception
- Elicitation
- Analysis and Elaboration
- Prioritization, Negotiation
- Specification
- Validation

□ Requirements Management

REQUIREMENT MANAGEMENT

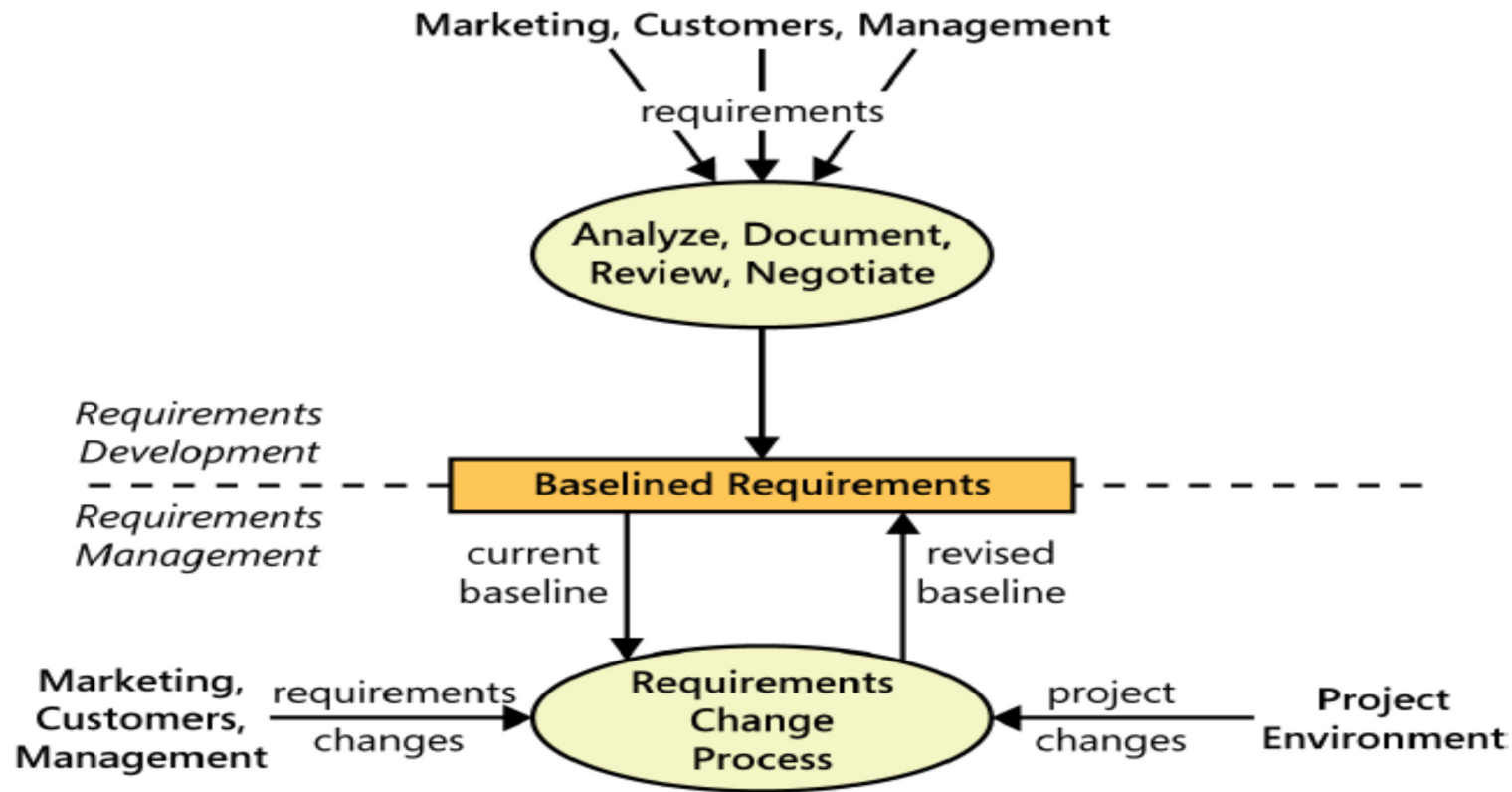


FIGURE 1-5 The boundary between requirements development and requirements management.

REQUIREMENT INCEPTION

□ **Inception**—ask a set of questions that establish:

- basic understanding of the problem
- the people who want a solution (identify the stakeholder)
- the nature of the solution that is desired
- the effectiveness of preliminary communication and collaboration between the customer and the developer
- what will be the economic benefit of a successful solution

REQUIREMENT ELICITATION

- **Elicitation**—elicit requirements from all stakeholders (use CRC card)
 - Interviewing related stakeholder with pre-determined questionnaire
 - meetings are conducted and attended by both software engineers and customers
 - Observation and ethnography
 - a "definition mechanism" (can be work sheets, flip charts, or wall stickers or an electronic bulletin board, chat room or virtual forum) is used in collecting requirements
 - the goal is:
 - to identify the problem
 - propose elements of the solution
 - specify a preliminary set of solution requirements

REQUIREMENTS ANALYSIS AND ELABORATION

Building the Analysis Model:

☐ Scenario-based elements

- Functional—processing narratives for software functions
- Use-case—descriptions of the interaction between an “actor” and the system

☐ Class-based elements

- Implied by scenarios

☐ Behavioral elements

- State diagram

☐ Flow-oriented elements

- Data flow diagram, Sequence diagram, Activity Diagram

REQUIREMENTS NEGOTIATION & PRIORITY

- ❑ Identify the key stakeholders
 - These are the people who will be involved in the negotiation
- ❑ Determine each of the stakeholders “win conditions”
 - Win conditions are not always obvious
- ❑ Negotiate/Prioritization
 - Work toward a set of requirements that lead to “win-win”

REQUIREMENT VALIDATION

- Is each requirement **consistent with the overall objective** for the system/product?
- Have all requirements been specified at the **proper level** of abstraction?
- Is the **requirement really necessary** or does it represent an add-on feature that may not be essential to the objective of the system?
- Is each requirement **unambiguous**?
- Do any requirements **conflict** with other requirements?
- Is each requirement **achievable in the technical environment** that will house the system or product?
- Is each requirement **testable, once implemented**?

REQUIREMENTS DEVELOPMENT PROCESS FRAMEWORK

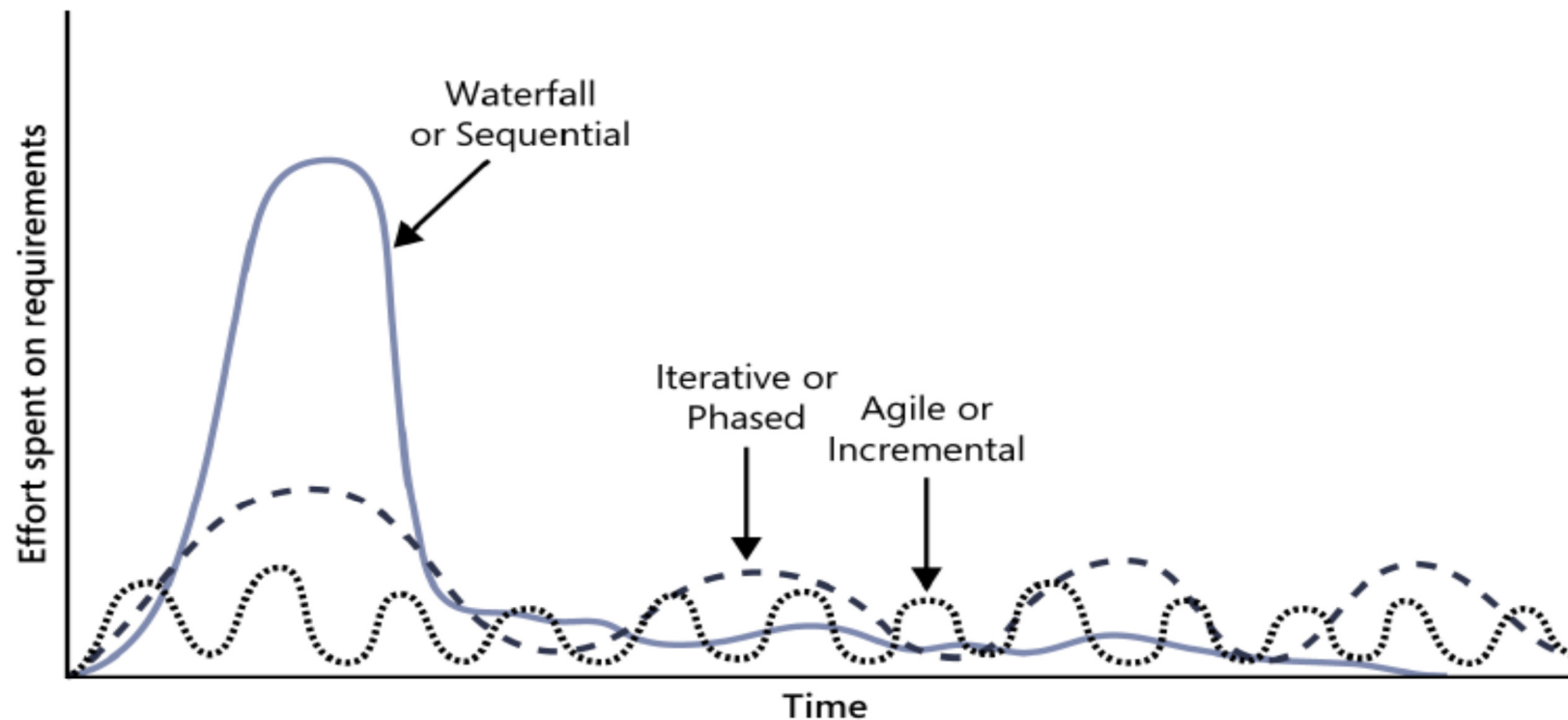


FIGURE 3-3 The distribution of requirements development effort over time varies for projects that follow different development life cycles.

REASONS BEHIND BAD REQUIREMENTS

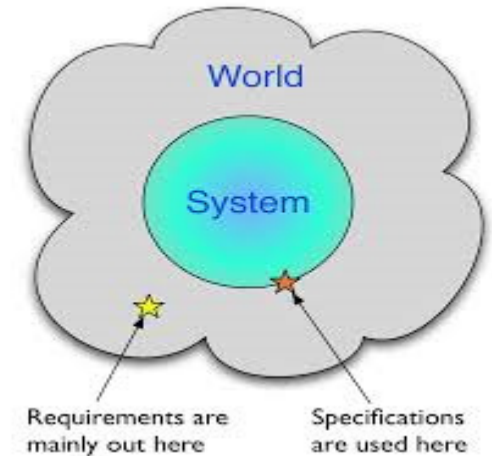
- Insufficient user involvement
- Inaccurate planning
- Creeping (gradually increase, changing) user requirements
- Ambiguous requirements (e.g., response to a request as soon as possible)
- Gold plating (extra functionality beyond the specification, MoSCoW)
- Overlooked stakeholders

BENEFITS OF A HIGH-QUALITY REQUIREMENTS PROCESS

- Fewer defects in requirements and in the delivered product
- Reduced development rework
- Faster development and delivery
- Fewer unnecessary and unused features
- Lower enhancement costs
- Fewer miscommunications
- Reduced scope creep
- Reduced project chaos
- Higher customer and team member satisfaction
- Products that do what they're supposed to do

REQUIREMENT VS SPECIFICATION

- **Requirement** refers the business need from the perspective from business user whereas the **specification** defines those requirements from system perspective. Requirement document what is needed whereas specifications document how to achieve the requirements.
- The **requirement** represents the problem or need whereas the **specification** provides the solution to that problem/ need.
- **Requirement** refers the business need from the perspective from business user whereas the **specification** defines those requirements from system perspective. Requirement document what is needed whereas specifications document how to achieve the requirements.
- The **requirement** represents the problem or need whereas the **specification** provides the solution to that problem/ need.
- The **requirement** is gathered from business user/ stakeholders whereas the **specification** is provided by technical team keeping requirements in mind.



REQUIREMENT VS SPECIFICATION

- The input for **requirement** is the business users whereas the input for **specification** is requirement document, business users and technical team.
- Looking at the difference between user requirements and system specifications in the ATM example, we know that swiping the card and prompting for a PIN are requirements, while reading the card details and a 4-digit PIN are specifications.

REFERENCES

- R.S. Pressman & Associates, Inc. (2010). *Software Engineering: A Practitioner's Approach*.
- Kelly, J. C., Sherif, J. S., & Hops, J. (1992). An analysis of defect densities found during software inspections. *Journal of Systems and Software*, 17(2), 111-117.
- Bhandari, I., Halliday, M. J., Chaar, J., Chillarege, R., Jones, K., Atkinson, J. S., & Yonezawa, M. (1994). In-process improvement through defect data interpretation. *IBM Systems Journal*, 33(1), 182-214.
- Wiegers, K., & Beatty, J. (2013). *Software requirements*. Pearson Education.
- <http://www.cs.ccsu.edu/~stan/classes/CS530/notes14/04-Requirements.html>