



Software Engineering Concepts

Introduction to Requirements & Requirement Engineering

Farrukh Zeshan Ahmad



Scope Management

- Scope refers to all the work involved in creating the products of the project and the processes used to create them
- How the project scope will be defined, validated, and controlled. The key benefit of this process is that it provides guidance and direction on how scope will be managed throughout the project.



Scope Management

- Identify Stakeholders
- Stakeholder Management Plan
- Objectives
- EEF
- OPA
- Expert Judgment
- Meetings
- Scope Management Plan
- Requirements Mang. Plan

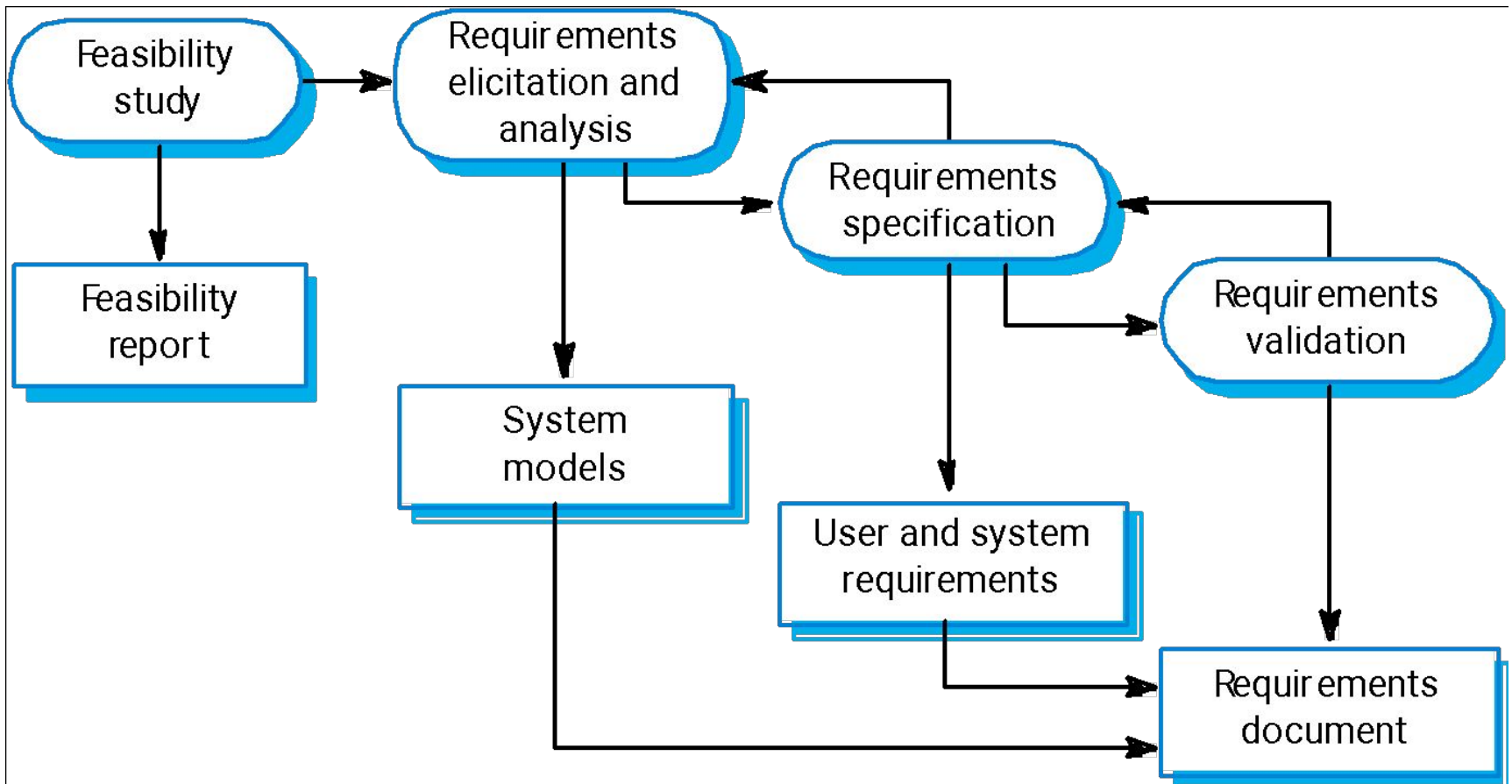


Understanding Requirements

- Requirements describe in detail what a software system is supposed to do, and they are the first step toward a solution.
- Requirement engineering is the disciplined application of proven principles, methods, tools, and notation to describe a proposed system's intended behavior and its associated constraints.



Requirements Engineering Process





- **Guidelines of Requirements Elicitation**

- Assess the business and technical feasibility for the proposed system
- Identify the people who will help specify requirements.
- Define the technical environment (e.g. computing architecture, operating system, telecommunication needs) into which the system or product will be placed.



- **Guidelines of Requirements Elicitation**

- Identify “domain constraints” (i.e. characteristics of the business environment specific to the application domain) that limit the functionality or performance of the system or product to build
- Create usage scenarios of use cases to help customers/ users better identify key requirements.



Collect Requirements

- Collect Requirements is the process of determining, documenting, and managing stakeholder needs and requirements to meet project objectives.
- The key benefit of this process is that it provides the basis for defining and managing the project scope including product scope.



Collect Requirements

- Scope Mng. Plan
- Req. Mng. Plan
- Stakeholders Mng. Plan
- PC
- Stakeholder Register.



Collect Requirements

- Interview
- Focus groups
- Facilitated workshops
- Group creativity techniques
- Group decision making techniques.
- Questionnaires and surveys
- Observations
- Prototypes
- Benchmarking
- Context diagrams
- Document analysis



Viewpoint-oriented analysis

- **Stakeholders** represent **different ways of looking** at a problem or problem viewpoints
 - different types of stakeholders
 - different views among stakeholders of same type
- This multi-perspective analysis is important as there is **no single correct way** to analyse system requirements



Collect Requirements

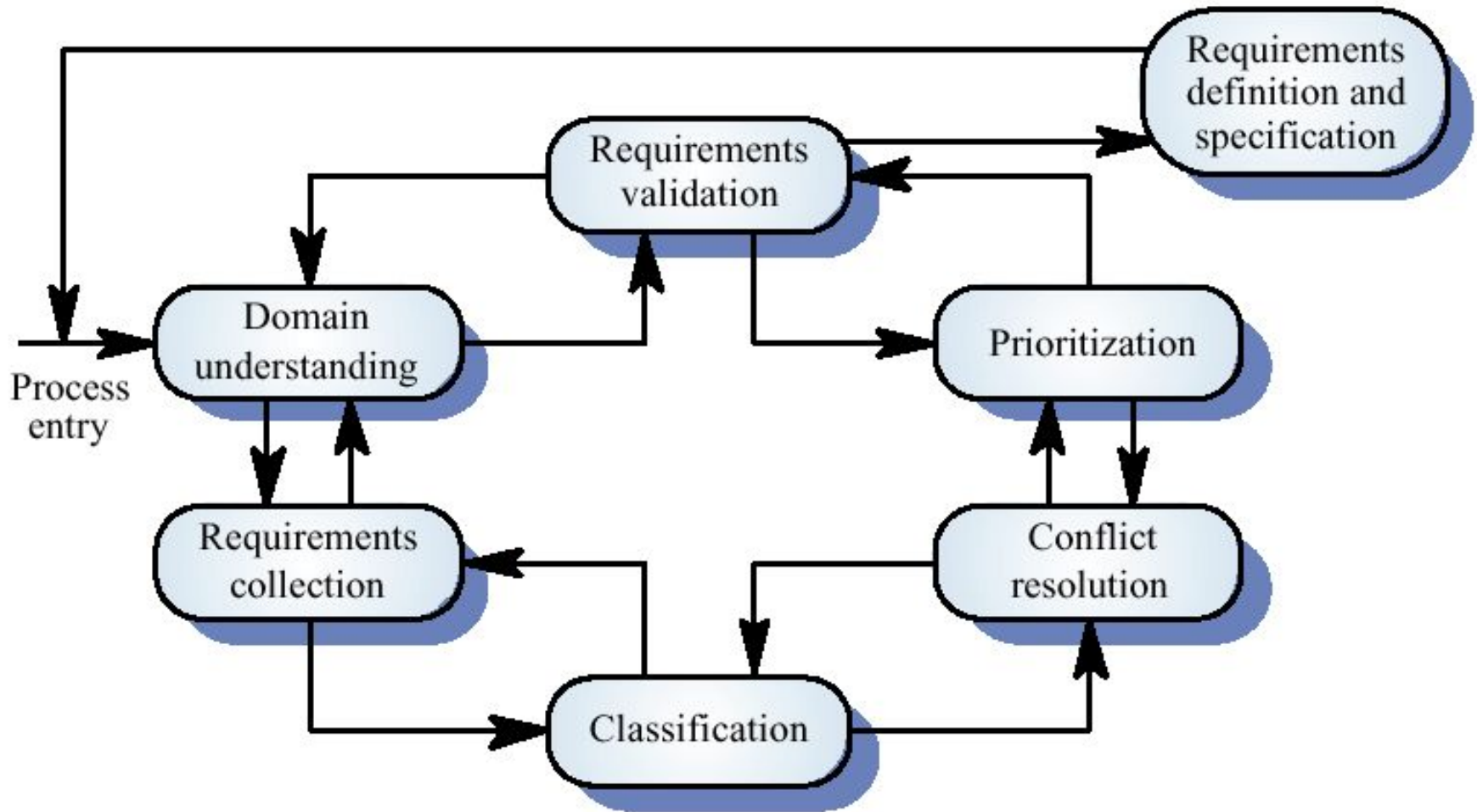
- Requirements documentation
- Requirements traceability matrix



Requirement Analysis

- To determine whether the stated requirements are clear, complete, consistent and unambiguous.

Requirements Analysis





Classification of Requirements

- Requirements can be classified as
 - a) Functional Requirements and
 - b) Non-Functional Requirements.
- “Non-functional” do not function or do not serve any function directly but they are serving a useful purpose in the software, They indirectly, perhaps, assist in the better functioning of business processes.

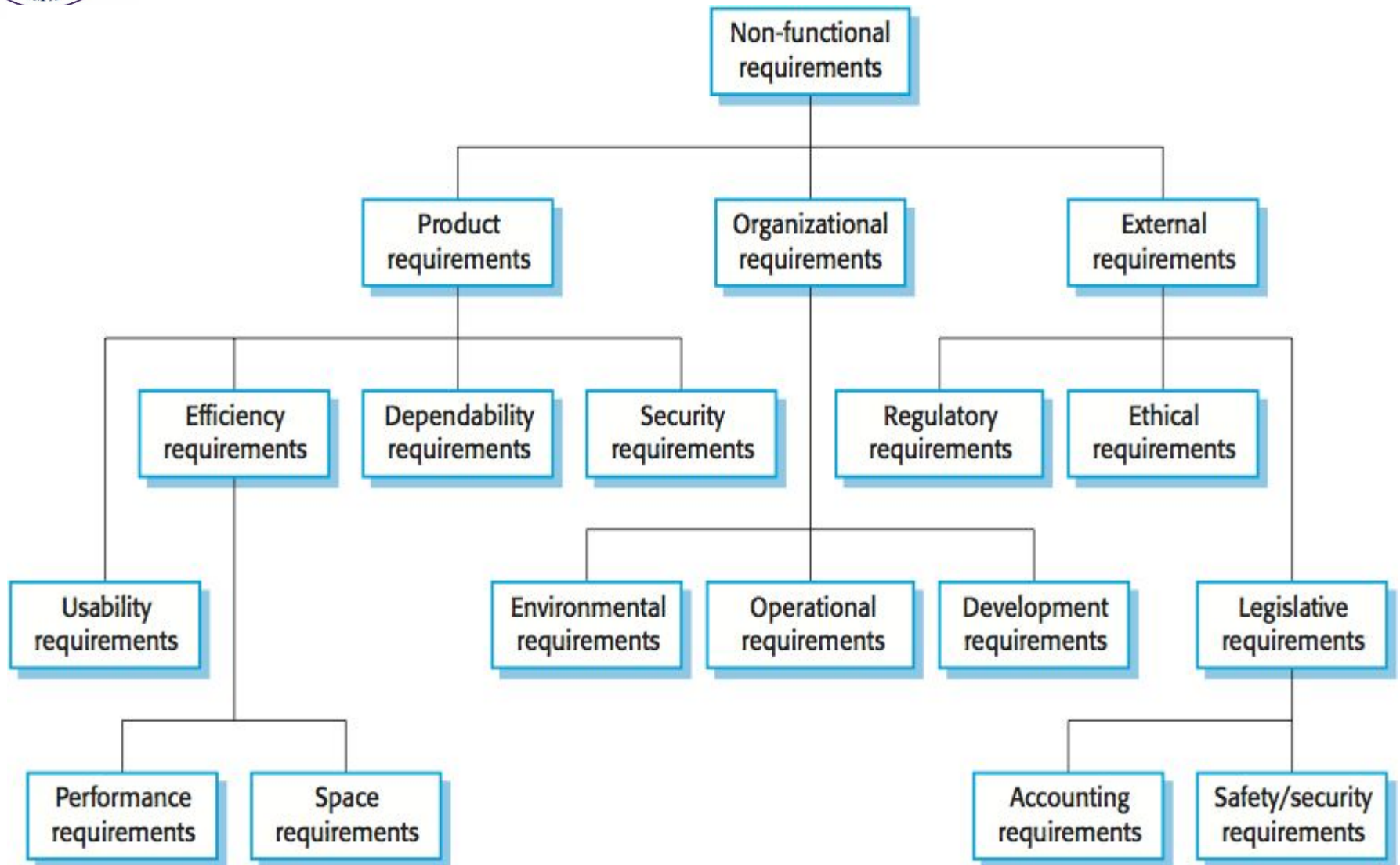


Functional Requirements

- What inputs the system should accept
- What outputs the system should produce
- What data the system should store that other systems might use
- What computations the system should perform



Non-Functional Requirements





Non-Functional Requirements

- Response time • Throughput • Resource usage • Reliability • Availability • Recovery from failure • Allowances for maintainability and enhancement • Allowances for reusability



Nonfunctional Requirement Metrics

Property	Measure
Speed	Processed transactions/second User/event response time
Ease of use	Training time
Reliability	Mean time to failure Probability of unavailability Rate of failure occurrence Availability
Robustness	Time to restart after failure Percentage of events causing failure Probability of data corruption on failure



Define Scope

- Define Scope
- Product analysis
- Alternative generation
- Facilitated workshops

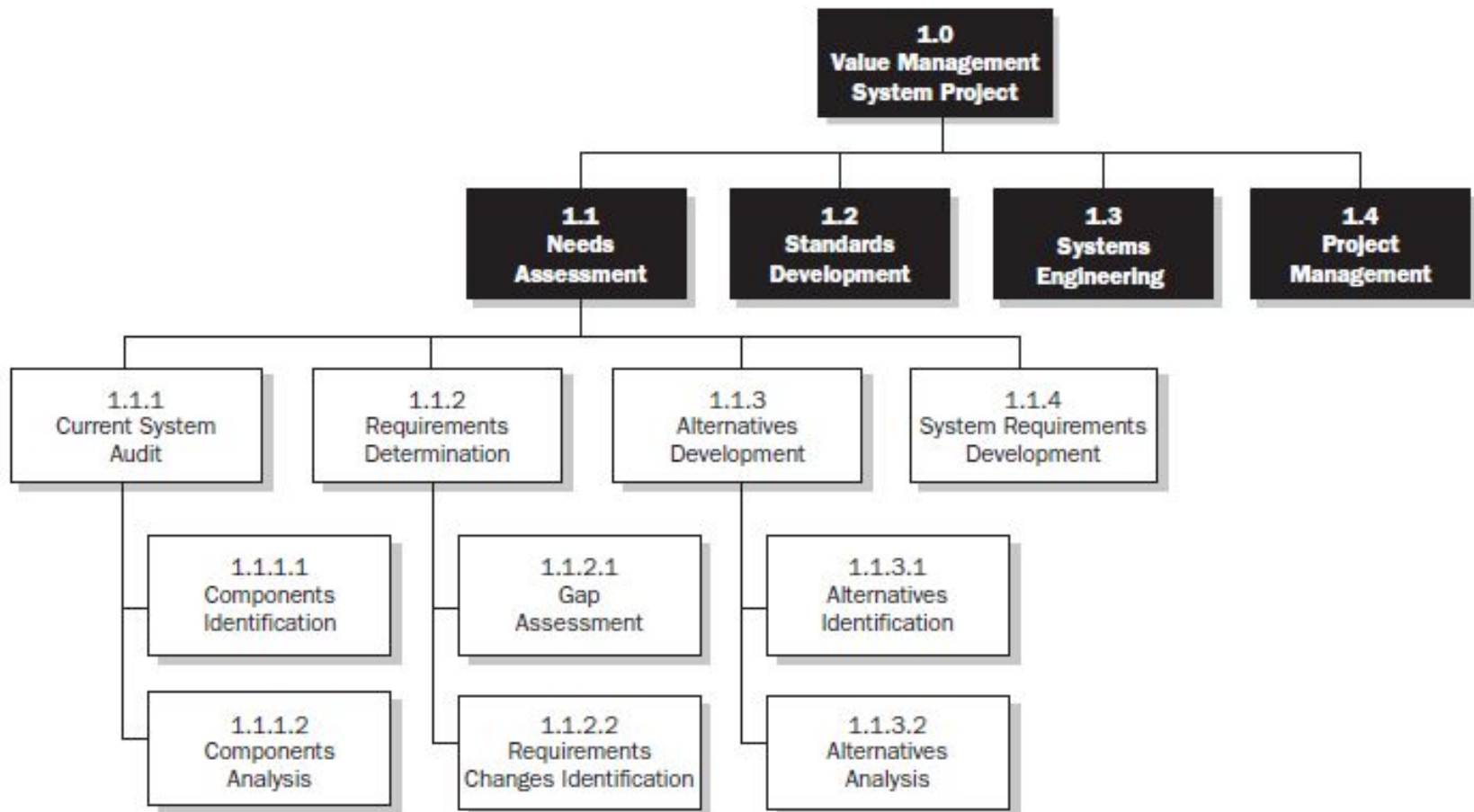


Create Work Breakdown Structure

- A WBS is a deliverable-oriented grouping of the work involved in a project that defines the total scope of the project
- WBS is a foundation document that provides the basis for planning and managing project schedules, costs, resources, and changes
- Decomposition is subdividing project deliverables into smaller pieces
- A work package is a task at the lowest level of the WBS



Work Breakdown Structure





Validate Scope

Inspection

Group decision-making techniques



Validation & Verification

- Validation (& Verification), is the process of checking whether the requirements, as identified, do not contradict the expectations about the system of various stakeholders and do not contradict each other.



Validation & Verification

- Validation: “Am I building the right product?” checking a work product against higher-level work products or authorities that frame this particular product.
 - Requirements are validated by stakeholders
- Verification: “Am I building the product right?” checking a work product against some standards and conditions imposed on this type of product and the process of its development.
 - Requirements are verified by the analysts mainly

[illegible]



Traceability Matrix Purpose

- To check that all requirements are covered by the use cases
- To check that none of the use cases is introduced without a reason (i.e., created not in response to any requirement)
- To prioritize the work on use cases



Use cases (Descriptions)

- *Use cases*, provide a description of how the system will be used.
- Use-cases are scenario-based technique that identifies how the actors will interact with the system.
- A set of use cases should describe all possible interactions with the system.
- Sequence diagrams may be used to add detail to use-cases by showing the sequence of event processing in the system.



Use case Questions

A use case should answer the following question:

- Who is the primary actor, the secondary actor(s)?
- What are the actor's goals?
- What preconditions should exist before the story begins?
- What main tasks or functions are performed by the actor?
- What exceptions might be considered as the story is described?



Use case Questions

- What variations in the actor's interaction are possible?
- What system information will the actor acquire, produce, or change?
- Will the actor have to inform the system about changes in the external environment?
- What information does the actor desire from the system?
- Does the actor wish to be informed about unexpected changes?



<Use case Id: Delete Information>

Priority	3
Actors:	User
Use Case Summary	Deleting information allows the user to permanently remove information from the system. Deleting information is only possible when the information has not been used in the system.
Pre-condition:	Information was previously saved to the system and a user needs to permanently delete the information.
Scenarios	
Step#	Scenario
1.	The use case starts when the user wants to delete an entire set of information such as a user, commission plan, or group.
2.	The user selects the set of information that he/she would like to delete and directs the system to delete the information. - Exception 1, 2
3.	The system responds by asking the user to confirm deleting the information.
4.	The user confirms deletion.
5.	Alternative Path: Cancel Action
6.	A system responds by deleting the information and notifying the user that the information was deleted from the system.
7.	Uses: Record Transaction



Alternate Scenarios: The user does not confirm Deletion

1a: If the user does not confirm deletion, the information does not delete.

1b: Uses: Cancel Action

Exceptions / Alerts

1. The system will not allow a user to delete information that is being used in the system.
2. The system will not allow a user to delete another user that has subordinates.

Post Conditions

Step#	Description
	The information is no longer available anywhere in the system.

Use Case Cross References

Includes	Record Transactions, Cancel Action
Extends	None



<UC-2: Course Registration>

Priority	3
Actors:	Student
Use Case Summary	Course registration information allows the user to add information about some object into the system.
Pre-condition:	Student must be logged on to the system.
Scenario	<ol style="list-style-type: none">1. Student selects “Course Registration” tab.2. Student selects session, course, and lecture and lab section as desired from drop down menu.3. Student selects “Look Up” button.4. Student pressed “Add” button.5. System displays the course indicating that the student has successfully registered in the course.



Alternate Scenarios: The user does not confirm Registration

1a: If the user does not confirm registration, the information does not recorded.

1b: Uses: Cancel Action

Exceptions / Alerts

1. The system will not allow a user to add information twice.

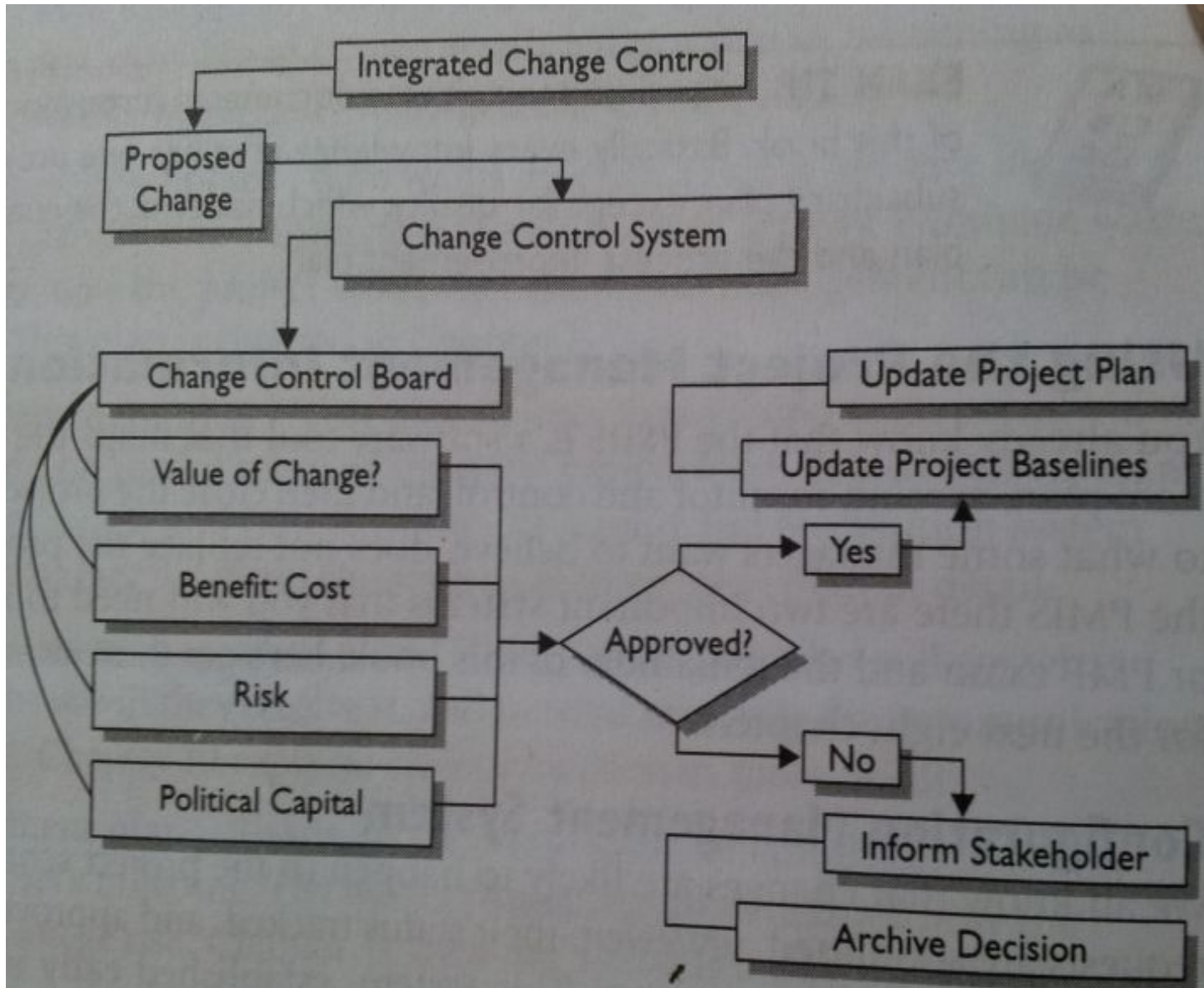
Post Conditions

The information database of the system will be updated.

Use Case Cross References

Includes	None
Extends	None

Requirements Change Management





Software Requirements Specification

- A Software Requirements Specification (SRS) – a requirements specification for a software system – is a complete description of the behavior of a system to be developed. It includes a set of use cases that describe all the interactions the users will have with the software.



Some Risks From Inadequate Requirement Process

- Insufficient user involvement leads to unacceptable products.
- Creeping user requirements contribute to overruns and degrade product quality.
- Ambiguous requirements lead to ill-spent time and rework.
- Gold-plating by developers and users adds unnecessary features.
- Minimal specifications lead to missing key requirements.
- Overlooking the needs of certain user classes (stake holders) leads to dissatisfied customers.
- Incompletely defined requirements make accurate project planning and tracking impossible.



Ambiguous requirements

- Ambiguity – arising from natural language
 - Sommerville – no output is better than wrong output.
 - Rooko mut jane do
- Leads to different expectations
- Waste of time and effort



Contradiction

- System must monitor all temperatures in a chemical reactor.
- System should only monitor and log temperatures below -20°C and above 400°C .



Thank You



Ref.

- <https://cs.ccsu.edu/~stan/classes/CS410/notes16/04-Requirements.html>
- PMBok

Code Complete

Roger s. pressman.