

B. Computer Science (SE) (Hons.)

CSEB233: Fundamentals of Software Engineering

Understanding
Requirements Engineering



**UNIVERSITI
TENAGA
NASIONAL**



Objectives

- Discuss the concept of requirements and the types of requirements
- Explain what Requirements Engineering is, its process, and its importance to product development projects
- Explain and relate requirements elicitation, requirements analysis and negotiation, requirements specification, requirements verification and validation, and requirements management activities
- Describe various methods, approaches, and techniques for performing and supporting the Requirements Engineering process



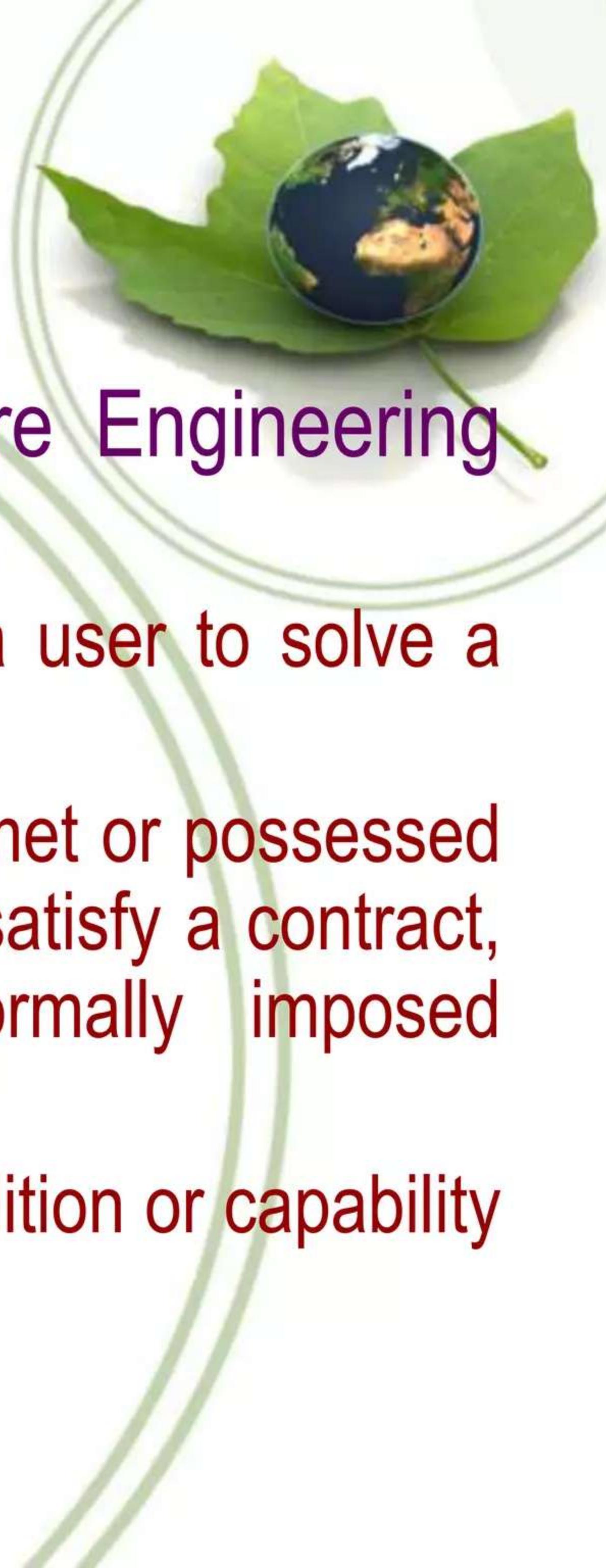
Understanding Requirements Engineering

UNIVERSITI
TENAGA
NASIONAL

**Concept and Types of
Requirements**

What are ‘Requirements’?

- IEEE Standard Glossary of Software Engineering Terminology (IEEE, 1990):
 - “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 (1) and (2).”



Types of Requirements

- Complementary requirements
 - Functional Requirements (FR)
 - Non-functional Requirements (NFR)



Functional Requirements

- IEEE Standard Glossary of Software Engineering Terminology (1990) define FR as:
 - “a requirement that specifies a function that a system or system component must be able to perform”
- Function is defined as:
 - “a defined objective or characteristic action of a system or component”
 - For example, a system may have inventory control as its primary function
- Functional requirements relate to the actions (such as calculate, retrieve, display) that the system or system component must carry out in order to satisfy the reason for its existence

(Robertson & Robertson, 1999)

Functional Requirements

- Describe the services a system or component of a system should perform
- Tell you and your users how the system should react to certain inputs
- Describe how the system should and/or should not behave in particular situations
- Must not include quality statement such as 'fast', 'efficient', 'usable', 'reliable', and etc.
- Are important for the developers to use them to develop the system as expected by the customers



Non-functional Requirements

- A requirement that specifies quality characteristics /attributes of the software and constraints of the software to be developed and/or process to develop the software

Non-functional Requirements

- Classifications of NFRs:
 - Quality characteristics/attributes (Qualities) - usually associated with the system as a whole and not to individual function - will affect degree of user satisfaction
 - Product - specify that the delivered product must behave in a particular way e.g. execution speed, reliability, etc.
 - Organizational - a consequence of organizational policies and procedures, e.g. process standards used, implementation requirements, etc.
 - External - arise from factors which are external to the system and its development process e.g. interoperability requirements, legislative requirements, etc.
 - Constraints
 - non-negotiable and are off-limits during design trade-offs

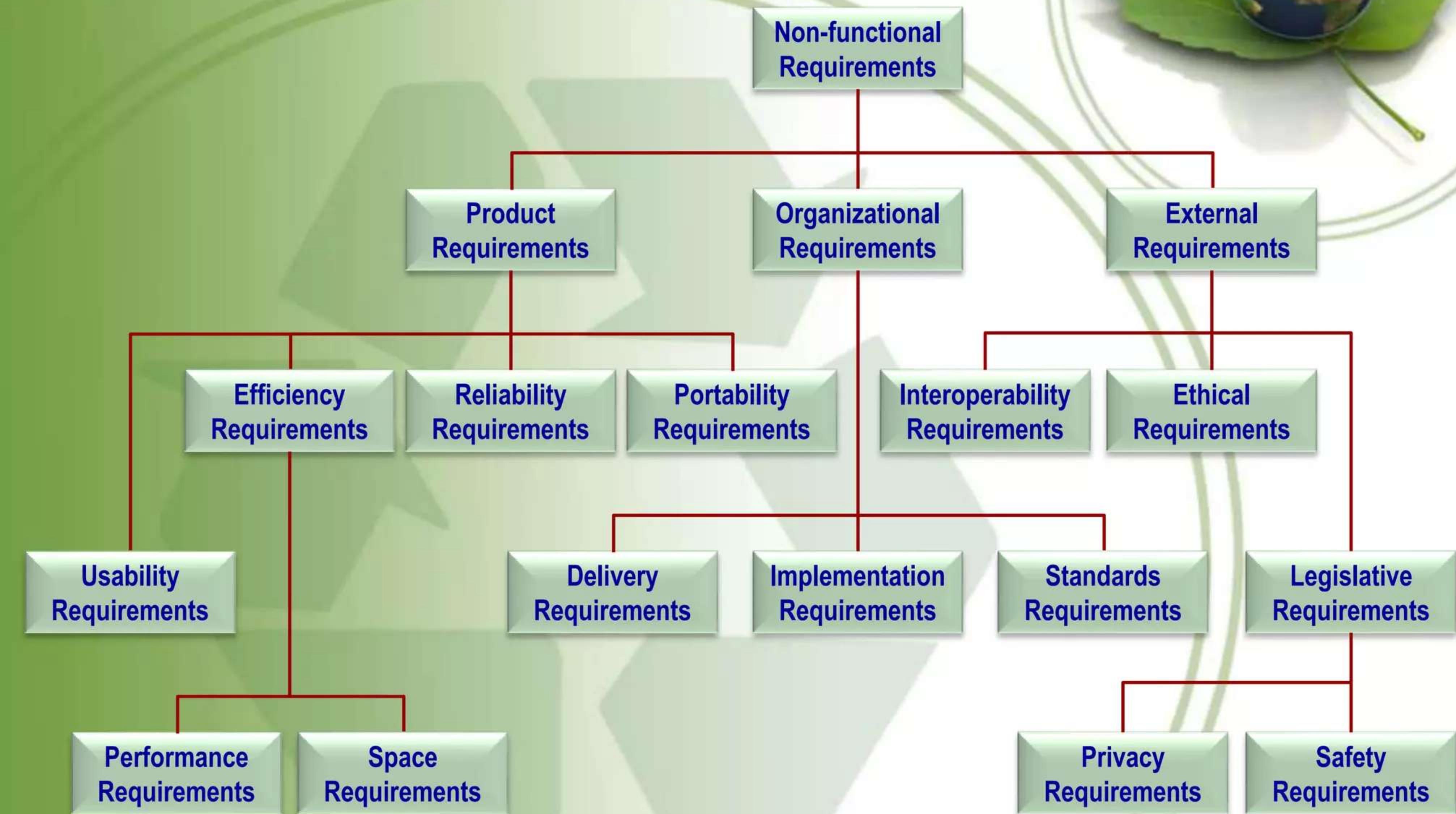


Non-functional Requirements

- Example quality attributes:
 - Reliability; Security; Performance
 - Ethical; Safety; Delivery
- Example constraints:
 - Skill-set of the developers
 - Target operating environment
 - Programming language to develop the system
 - Software process that should be followed
 - Design that must be produced with a particular CASE tool



NFRs: Qualities



Adopted from Sommerville (2007)

Understanding Requirements Engineering

UNIVERSITI
TENAGA
NASIONAL

**What is Requirements
Engineering?**

What is Requirements Engineering (RE)?

- "... a relatively new term which has been invented to cover all of the activities involved in discovering, documenting, and maintaining a set of requirements for a computer-based system.
- The use of the term 'engineering' implies that systematic and repeatable techniques should be used to ensure that systems are complete, consistent, relevant, etc".

Sommerville & Sawyer (1997)

- "the process of developing a requirements specification"

Pohl (1996)

- "the broad spectrum of tasks and techniques that lead to an understanding of requirements"

Pressman (2009)

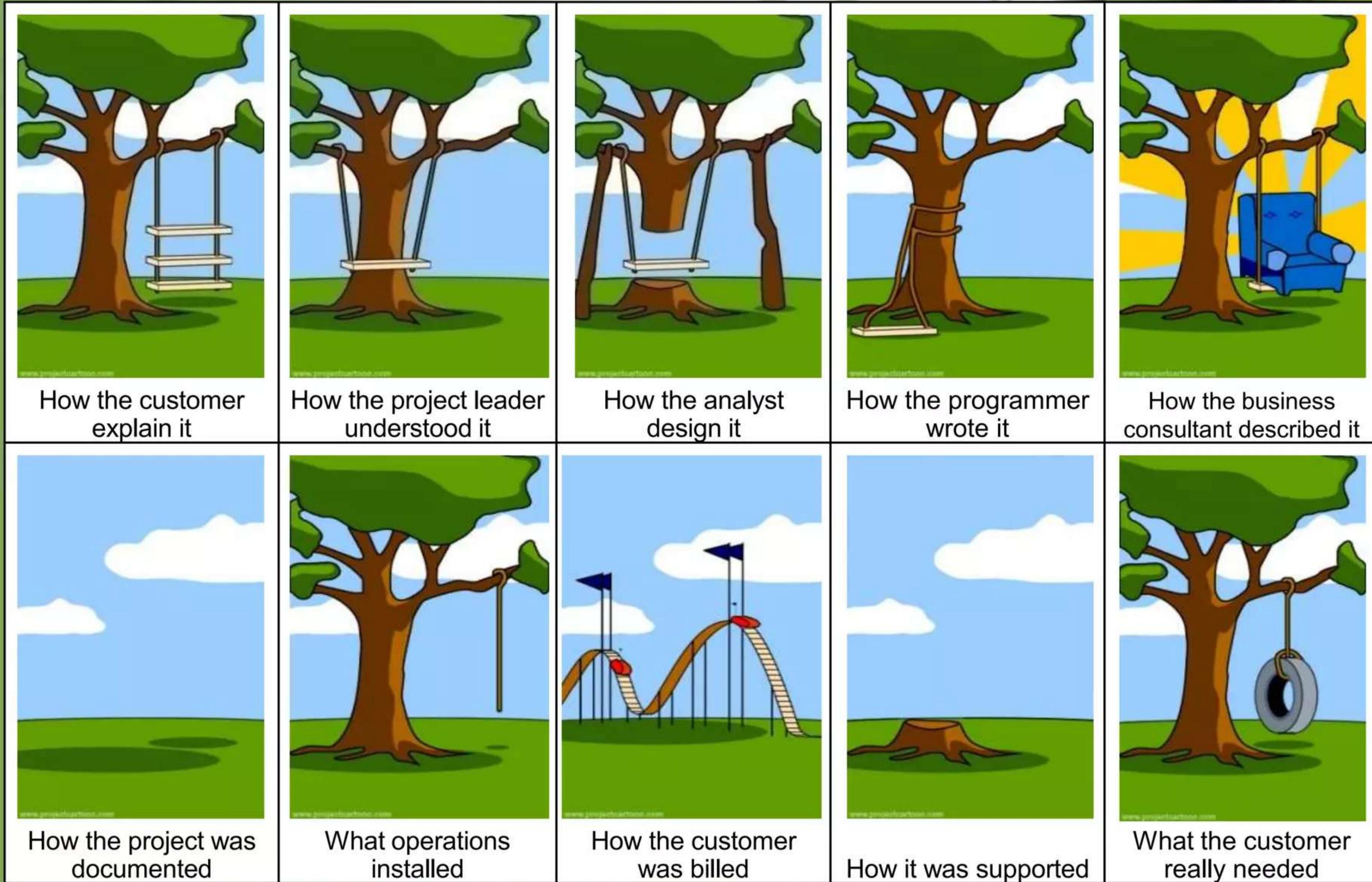
Why is RE important?

- RE process can influence the development cost, time, effort, and quality of the product
- RE process is an essential contributor to the overall quality of the software product
- “Incomplete requirements”, “changing requirements” are major causes of project failures
- Good RE practices contribute more than 42% towards the overall success of a project, while improper RE practices account for more than 43% of the reasons why projects are late or over budget

(CHAOS, 1995)

Importance of RE

(<http://www.projectcartoon.com>)

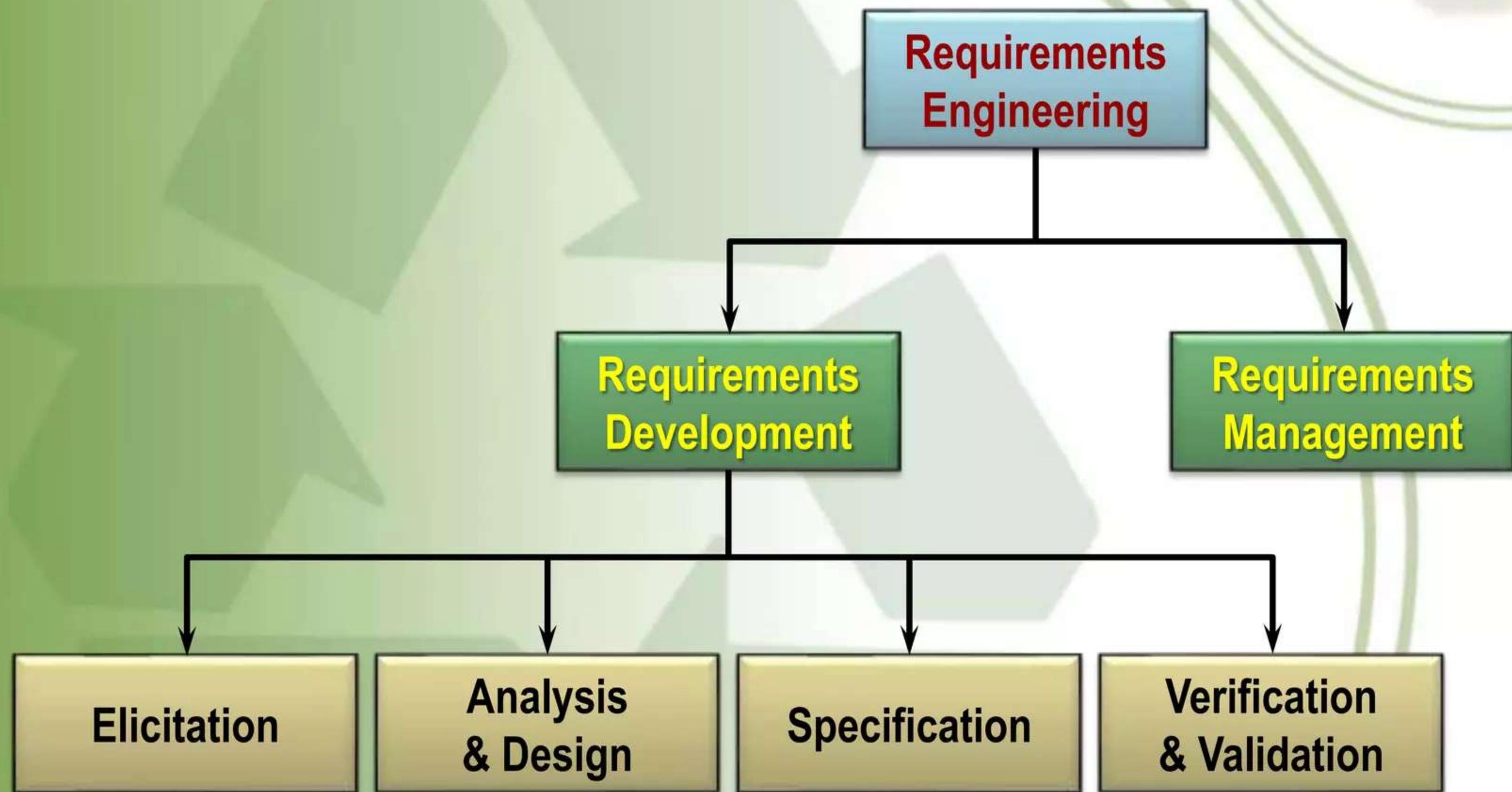


RE Process

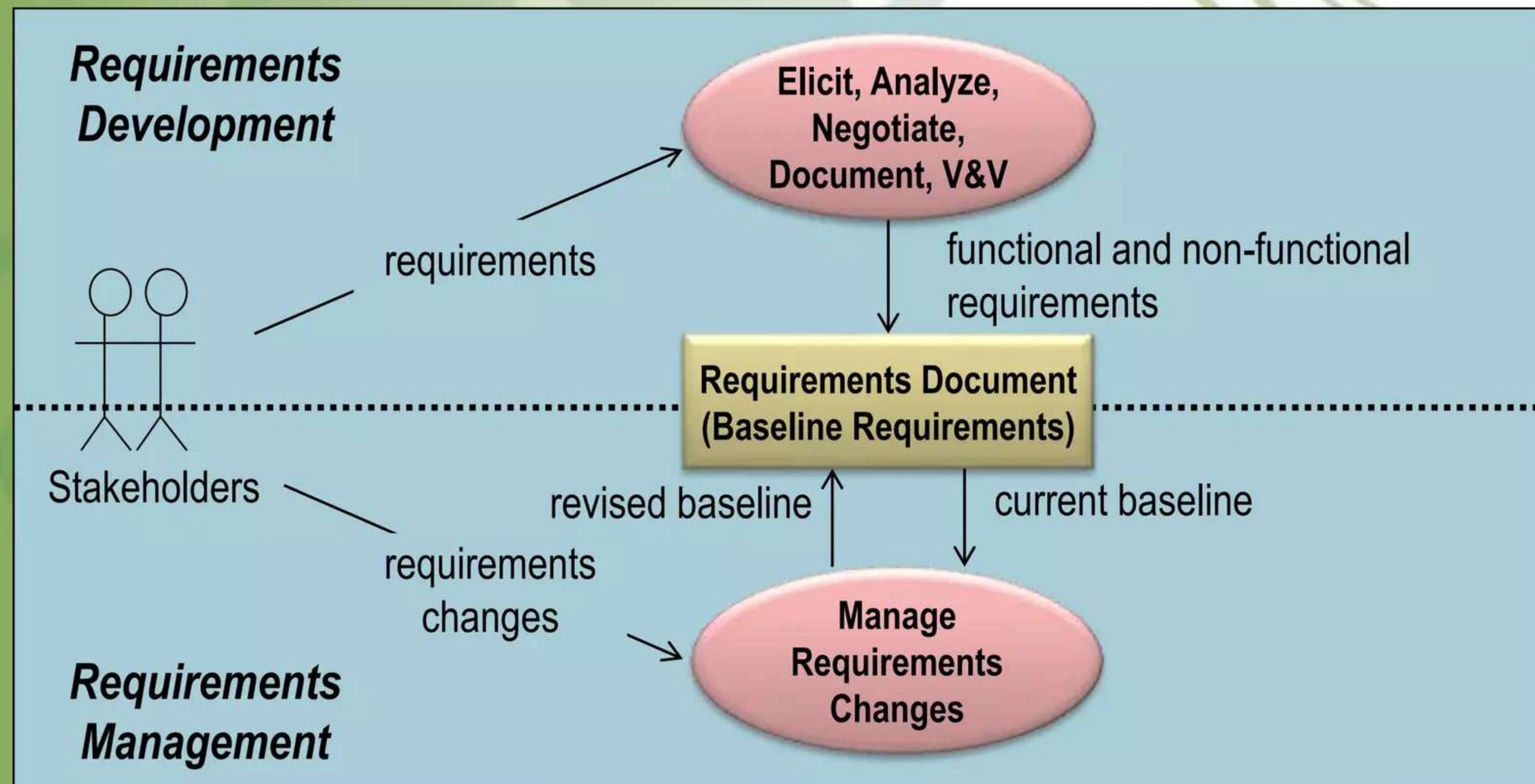
- RE process: “a structured set of activities which are followed to derive, validate and maintain a systems requirements document”
(Sommerville and Sawyer, 1997)
- Main tasks:
 - Requirements Elicitation
 - Requirements Analysis and Negotiation
 - Requirements Specification
 - Requirements Verification and Validation
 - Requirements Management



RE Process



The Boundary of RE Process



Understanding Requirements Engineering

UNIVERSITI
TENAGA
NASIONAL

Requirements Elicitation

The Groundwork

Inception

- May be performed via a project blastoff meeting to discover the following information:
 - Product purpose – are there any other products that are compatible or that cover similar domain or work areas?
 - Project sponsors – who will pay for the development project?
 - Clients – for commercial products, who is going to purchase the product?
 - Users – who is going to use the product?
 - Other stakeholders – who are the people that can influence the product or know about some of the requirements for the product?



The Groundwork

Inception

- Other sources of requirements – are there standards, system documentation, business process documentation, policy, regulation or legal documents that can be referred to?
- Constraints – are there any product and development organization constraints that the developers must give consideration? Have they been specified? How much is the project going to cost? How much time is available to complete the project and deliver the product? How much effort is needed to develop and deliver the product?
- Scope of the work – what are the boundaries of the product and the project?
- Relevant facts and assumptions – what does everybody need to know? What assumptions are applicable to the project?
- The risk – what are the main risks faced by the project?



1. Requirements Elicitation

- A critical task in RE
 - To discover requirements from the stakeholders
- Four components of requirements elicitation:
 - Understanding application domain – this is about knowledge of the general area where the system is applied.
 - Understanding problem to be solved – understand details of the problem where the software will be applied.
 - Understanding business processes in an organization – understand how systems interact and affect the different part of the business and how they contribute to overall business goals.
 - Understanding the needs and constraints of the stakeholders – understand the work processes that the system is intended to support, the ways in which the system is likely to be used, and restrictions on the degree of freedom we have in providing a solution.



Elicitation Work Products

- a statement of need and feasibility
- a bounded statement of scope for the system or product
- a list of customers, users, and other stakeholders who participated in requirements elicitation
- a description of the system's technical environment
- a list of requirements (preferably organized by function) and the domain constraints that apply to each
- a set of usage scenarios that provide insight into the use of the system or product under different operating conditions
- any prototypes developed to better define requirements



Why is it difficult to understand what the customer wants?

- Problems of scope
 - System/software boundary is ill-defined
 - Customers/users specify unnecessary technical detail that may confuse overall system/software objectives
- Problems of volatility
 - Requirements change over time

(Christel and Kang, 1992)

Why is it difficult to understand what the customer wants?

- Problems of Understanding
 - Customers/users not sure what they want
 - Poor understanding of the capabilities and limitations of their own computing environment
 - Don't have full understanding of the domain problems
 - Have trouble communicating need
 - Omit information that is believed to be "obvious"
 - Specify ambiguous requirements
 - "I want a user friendly interface in the XYZ system".
 - Specify conflicting requirements

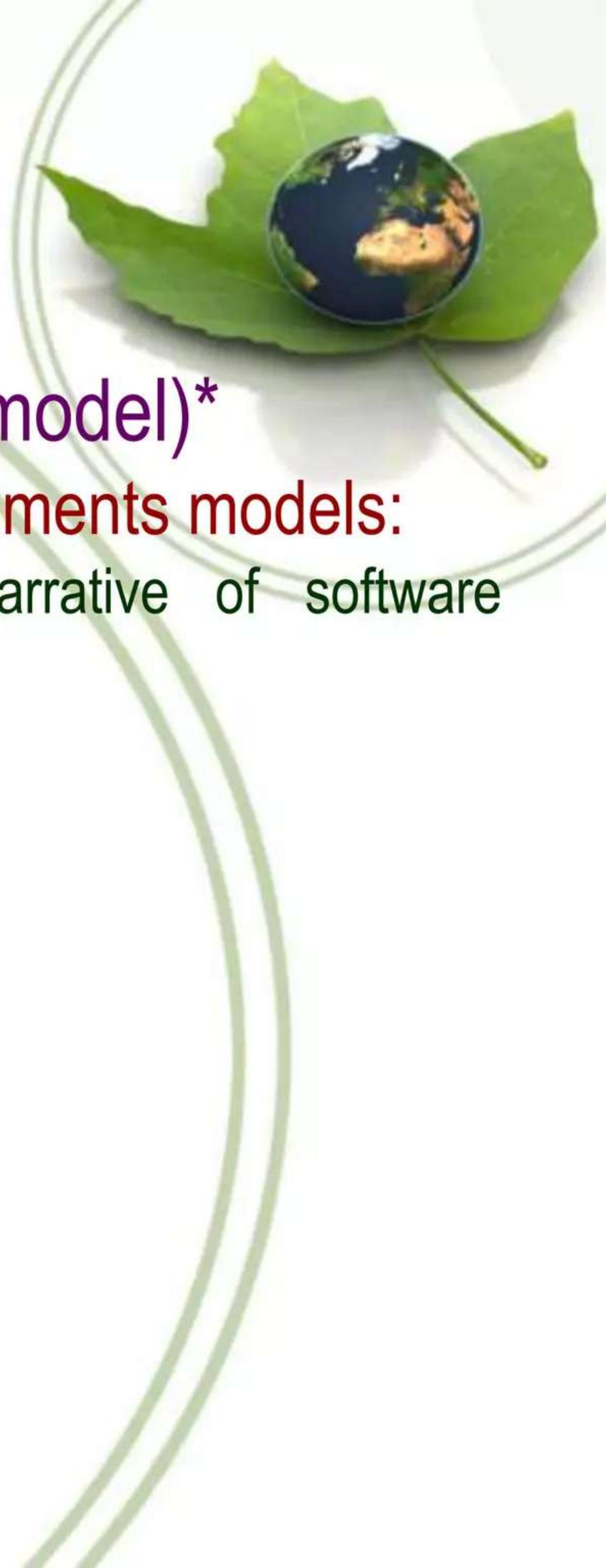
Elicitation Techniques



Traditional	Modern
<ol style="list-style-type: none">1. Interviews2. Questionnaires and Surveys3. Observations4. Analysis of Archival Document5. External Research	<ol style="list-style-type: none">1. Iterative Prototyping2. Joint Application Design (JAD)3. Rapid Application Development (RAD)4. Quality Function Deployment (QFD)*

2. Requirements Analysis & Negotiation

- Build analysis model (or requirements model)*
 - Generic elements common to most requirements models:
 - Scenario-based elements – processing narrative of software function
 - Class-based elements – implied by scenario
 - Behavioral elements – State Diagram
 - Flow-oriented elements - DFD
- Prioritizing Requirements
- Conflict and conflict resolution
- Negotiating requirements
- Assessing requirements risk



Prioritizing Requirements

- To decide which requirements have to be implemented and deliver first, which ones could be implemented in the subsequent deliveries, and which ones could be dropped etc.
- Must collaborate with the customers. Why?
 - You may not know which requirements are most important to customers, and
 - Customers may not be able to judge the cost and technical difficulty associated with specific requirements.



Techniques of Requirements Prioritization

- Prioritization scale - A common approach to prioritization is to group requirements into several priority categories.
 - E.g.: MoSCoW method (Coley Consulting, 2008)
 - M - MUST have this.
 - S - SHOULD have this if at all possible.
 - C - COULD have this if it does not affect anything else.
 - W - WON'T have this time but WOULD like in the future
- Quality Function Deployment
- Semi-quantitative Analytical Approach - the requirements' priority can be calculated once you have estimated the benefit, penalty, cost and risk for the negotiable requirements.



Conflict & Conflict Resolution

- In any set of requirements, there will always be conflicts, overlaps, and omissions
- Developer must anticipate these and plan requirements negotiation with all stakeholders to discuss and resolve the problems
- Requirements conflicts are inevitable because different stakeholders have different requirements and priorities
- Technique to identify conflicts and overlaps: interaction matrix

Negotiating Requirements

- A process of discussing the conflicts in the requirements and finding resolutions to the identified conflicts
- Theoretically, requirements negotiation should always be based on technical and organizational needs
 - Political factors and the people personalities have more influence to the negotiation process than the technical and organizational considerations
 - Handle the case well or risk the product to be rejected by certain group of stakeholders
- Effective technique to negotiate and resolve conflicts
 - Discussion sessions in meeting



Negotiating Guidelines

- Recognize that it's not a competition
 - Both parties have to feel they've won and must compromise
- Map out a strategy
- Listen actively
- Focus on the other party's interests
- Don't let it gets personal
- Be creative – think out of the box!
- Be ready to commit and move on



Requirements Risk

- Risk - generally defined as “a probability that some adverse circumstance will occur”
 - Identify risk
 - Technical, performance, safety and security, database, development process, schedule, external, legislative
 - Analyze risk
 - Assess probability of the risk from occurring and its possible effects
 - Mitigate risk

(Sommerville, 2007)



3. Requirements Specification

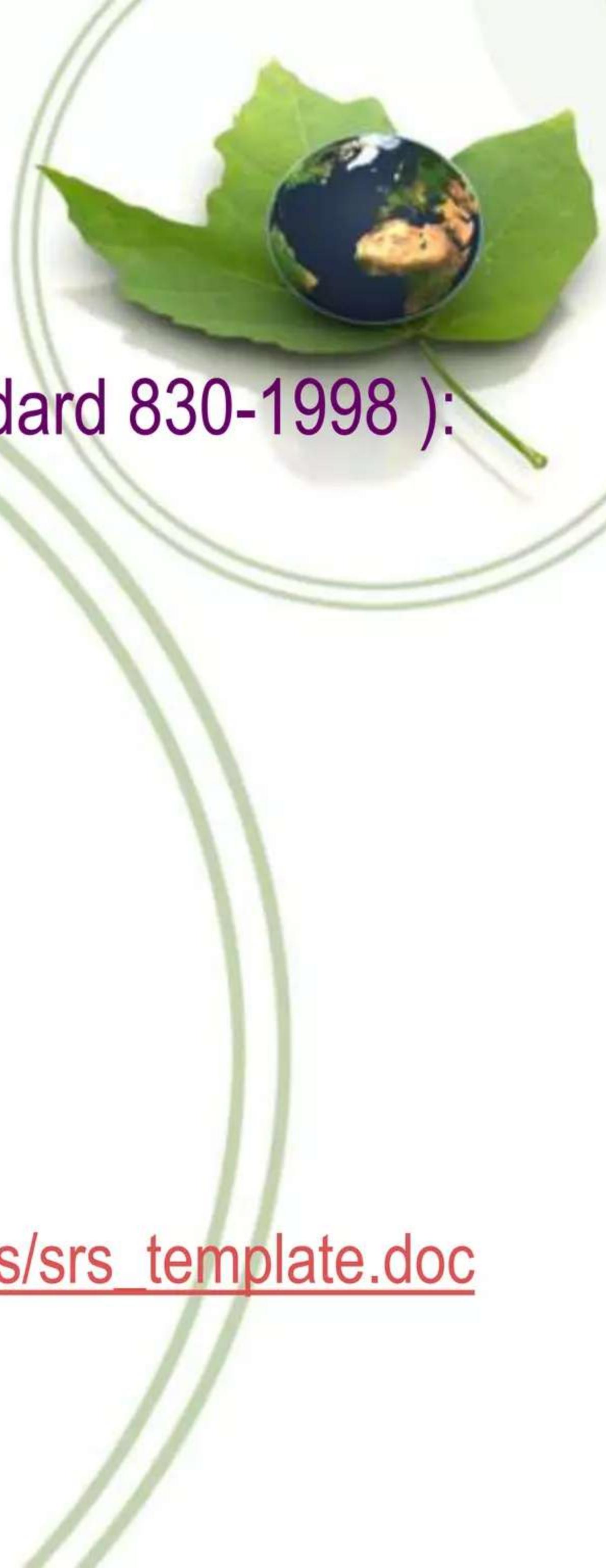
- To build the Software Requirements Specification (SRS) document
- An official document that consists of information that should guide the system developers such as designers, programmers, and engineers through the development work of the product

3. Requirements Specification

- Should involve technical writers
 - appropriate skills for gathering requirements, reviewing historic reports, writing formal documents and reports, and etc.
 - can better assess and plan documentation tasks
 - know how to determine the questions that are of concern to the customers and users regarding non-functional requirements like ease of use and usability

Software Requirements Specification (SRS)

- Attributes of a well-written SRS (IEEE Standard 830-1998):
 - Correct
 - Unambiguous
 - Complete
 - Consistent
 - Ranked for importance or stability
 - Verifiable
 - Testable
 - Traceable
- SRS template:
 - http://www.processimpact.com/process_assets/srs_template.doc



4. Requirements Verification & Validation

- Requirements verification is a process of checking that a product meets
 - its specification, and
 - the needs and expectations of the customer

(Sommerville, 2007)

Verification vs. Validation



Verification

A process for:

- Ensuring that the requirements have been defined.
- Determining that the requirements analysis has been correctly performed.
- Determining that the requirements provide all the information needed to develop the solution.
- Determining that the requirements are ready for formal review and validation by the customer and users.

Focuses on quality:

- Completeness,
- Correctness,
- Usability, etc.

Validation

A process for:

- Determining whether the requirements are accurate, correctly align to the needs of the customer, and have appropriate level of detail.

Focuses on:

- Supporting business goals/objectives,
- Aligning with business goals /objectives,
- Meeting stakeholders needs.

Verification and Validation Techniques

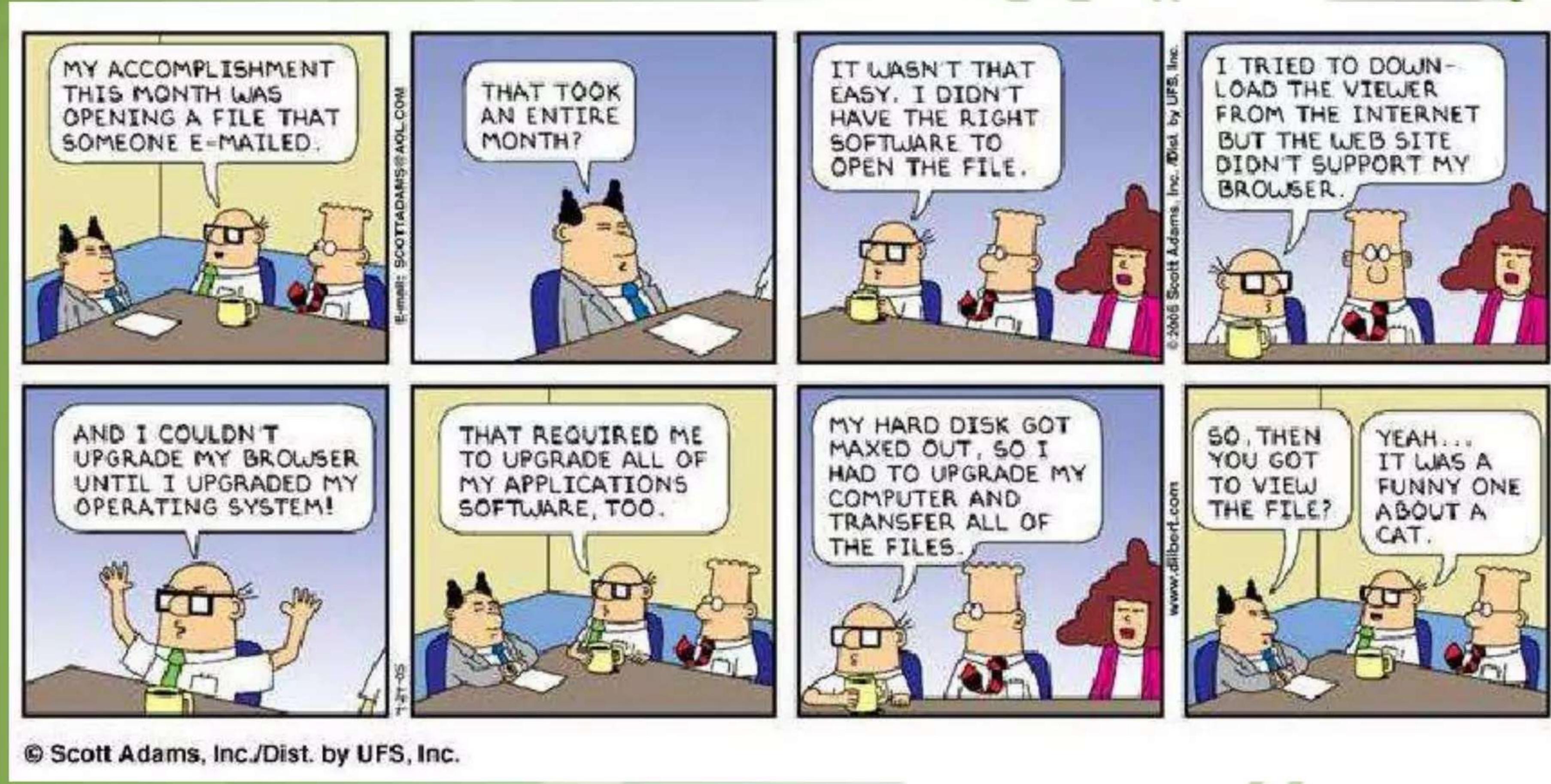
- Reviews the requirements specification
 - Desk-check
 - Walkthrough
 - Inspection
 - Checklist
- Prototyping
- Acceptance Tests



5. Requirements Management

- ‘Ripple Effect’ – one thing causes a series of other things to happen (e.g., Tsunami)
- Changes in requirements specified in a requirements document are inevitable and must be allowed
- However, even seemingly a minor change may unexpectedly require lot of work
- Requirements management:
 - Managing changes to requirements
 - Managing configuration of requirements and requirements document – version control
 - Maintaining requirements traceability
 - Tracking requirements status

Ripple Effect



Understanding Requirements Engineering

UNIVERSITI
TENAGA
NASIONAL

Supporting RE Process

Why Requirements Change?



Internal Factors	External Factors
<ol style="list-style-type: none">1. Failure to elicit the real requirements of the stakeholders2. Iterating from requirements to design creates new requirements3. The implementation team might encounter technical, schedule and/or cost problems in implementing a requirement4. RE process is iterative and must create a practical process to help manage changing requirements. Failure to create a practical requirements change management process will only cause rework and stress	<ol style="list-style-type: none">1. The problem we are trying to solve somehow change as a result of a changing economy, government regulations, consumer preferences etc.2. Customers and users understand better what they really require from a system or simply change their minds3. The customers' organization may change its structure, procedures and processes4. The external environment change, which create new constraints and opportunities

Managing Requirements Changes

- Formal change management is crucial to ensure that the requirements changes maintain the proposed system's support to the fundamental business goals
- To ensure a consistent approach to change management, organizations should define a set of change management policies and procedures

Managing Requirements Changes

- Basic policies:
 - The change management process – includes change management principles and guidelines, and activities of the change management process.
 - The change impact analysis – needed to avoid changes from causing overruns in project schedule and budget, or resulting negative impact on the product's quality.

Requirements Configuration Management

- Means detail recording and updating that have been applied to the requirements document, and providing version control, release management, and issue tracking.
- Benefits (Leffingwell and Widrig, 2003):
 - Prevents any unauthorised and potentially destructive changes to the requirements.
 - Preserves the revisions to requirements document.
 - Facilitates the retrieval and/or recreation of requirements document archives.
 - Prevents simultaneous updates of requirements documents.
 - Prevents conflicting and uncoordinated updates to different document at the same time.

Managing Requirements Traceability



- “The ability to describe and follow the life of a requirement, in both a forward and backward direction, i.e. from its origins, through its development and specification, to its subsequent deployment and use, and through periods of ongoing refinement and iteration in any of these phases”
(Gotel and Finkelstein, 1994)
- Technique: traceability matrix
 - to show the dependencies between requirements or links between requirements and other system documents

Traceability Matrix



Use case	Functional Requirements	Design Element	Code	Test Case
UC-28	catalog.query.sort	Class catalog	catalog.sort()	search.7 search.8
UC-29	catalog.query.import	Class catalog	catalog.import() catalog.validate()	search.8 search.13 search.14

Tracking Requirements Status

- Monitoring implementation status of each requirement
 - to ensure existing requirements are addressed and traceable throughout the development life cycle
- Tracking requirements status supports overall project status tracking.
 - e.g. proposed, approved, implemented, verified, deleted, rejected
- “If a project manager knows that 55% of the requirements allocated to the next release have been implemented and verified, 28% are implemented but not verified, and 17% are not yet fully implemented, then he or she has good insight into the project status”

(Wiegers, 1999a)

Summary

- The concept of requirements and types of requirements
- What Requirements Engineering is, its process, and the importance of them to product development projects in general
- What requirements elicitation, requirements analysis and negotiation, requirements specification, requirements verification and validation, and requirements management tasks are
- Various methods, approaches, and techniques for performing and supporting the RE process



THE END

Copyright © 2013

Mohd. Sharifuddin Ahmad, PhD
College of Information Technology

UNIVERSITI
TENAGA
NASIONAL

