Software Requirements Engineering

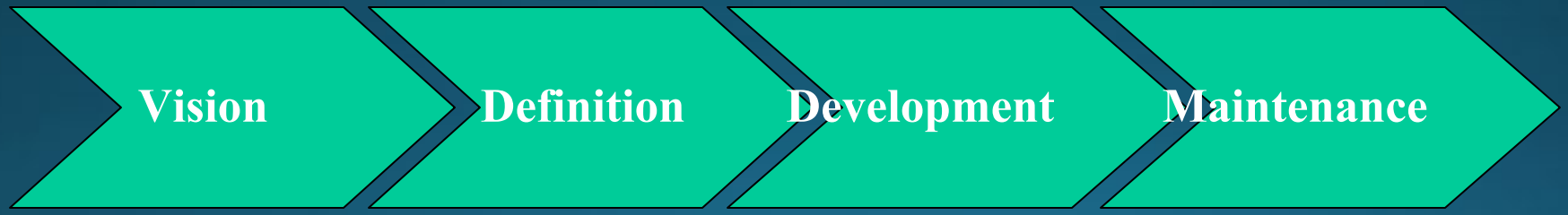
Introduction

(L#1-2)

Dr Sumaira Khan

Software Engineering Phases

1. Vision – focus on *why*
2. Definition – focus on *what*
3. Development – focus on *how*
4. Maintenance – focus on *change*



Requirements Engineering

- Requirement?
- Requirements Engineering?
- Importance of RE
- RE lifecycle
- Requirements Elicitation
- Writing Requirements
- Requirements Prioritization
- Requirements Validation
- Prototyping
- Requirements Management

Requirements Importance

- Buying a home?
- Buying a car?
- Buying a shoe?
- Buying a dress?
- Admit students in a program?
- Project partner?
- **Life partner?**

Why Requirements?

- Why waste time worrying about requirements?
- Why not save money by eliminating this unnecessary step?

Because - 1

- The later in the development cycle that a software error is detected, the more expensive it will be to repair
- Many errors remain latent and are not detected until well after the stage at which they are made
- There are many requirements errors being made

Because - 2

- Errors made in requirements specifications are typically incorrect facts, omissions, inconsistencies, and ambiguities
- Requirements errors can be detected

Potential Impact of Requirements Errors - 1

- The resulting software may not satisfy user's real needs
- Multiple interpretations of requirements may cause disagreements between customers and developers, wasting time and money, and perhaps resulting in lawsuits

Potential Impact of Requirements Errors - 2

- It may be (surely is) impossible to thoroughly test that the software meets its intended requirements
- Both time and money may be wasted building the wrong system
- Impact on humans

- Studies such as the **CHAOS report** [Johnson 2000] indicate that about half of the factors associated with project or product success are requirements related. Researchers have reported on studies showing that project success is directly tied to requirements quality [Kamata et al. 2007].
- Question is why is it still a relatively neglected topic in university training?
- It is quite rare, for example, that a new Computer Science (CS) university graduate might be asked to participate in the development of a compiler or operating system, yet nearly every graduate working in the industry will, sooner or later, be asked to participate in creating the requirements specifications for a product or service

Errors introduced during Requirements activities accounts 40-50% of all software defects found in software product (Davis 2005)

Requirements Engineering

- Customer Statement:
  - “I know you think you understand what I said, but what you don’t understand is what I said is not what I mean.”

Software Requirements?
(IEEE)

1 A condition or capability needed by user to solve a problem or achieve an objective.

2 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.

3 A documented representation of a condition or capability as in 1 or 2.

Software Requirements?

The statement of needs by a user that triggers the development of a program or system (Jones).

Software Requirement?

Requirements are ... A specification of what should be implemented. They are descriptions of how the system should **behave**, or of a system property or attribute. They may be a **constraint** on the development process of the system. (*Sommerville 1997)*

The Requirements Phase **(traditional)**

- Begins
  - There is a recognition that a problem exists and requires a solution
  - A new software idea arises
- Ends
  - With a *complete* description of the external behavior of the software to be built

Requirements Engineering - 1

- One of the most critical processes of system development
- Iterative system development - process repeated until requirements are of acceptable quality

Requirements Engineering - 2

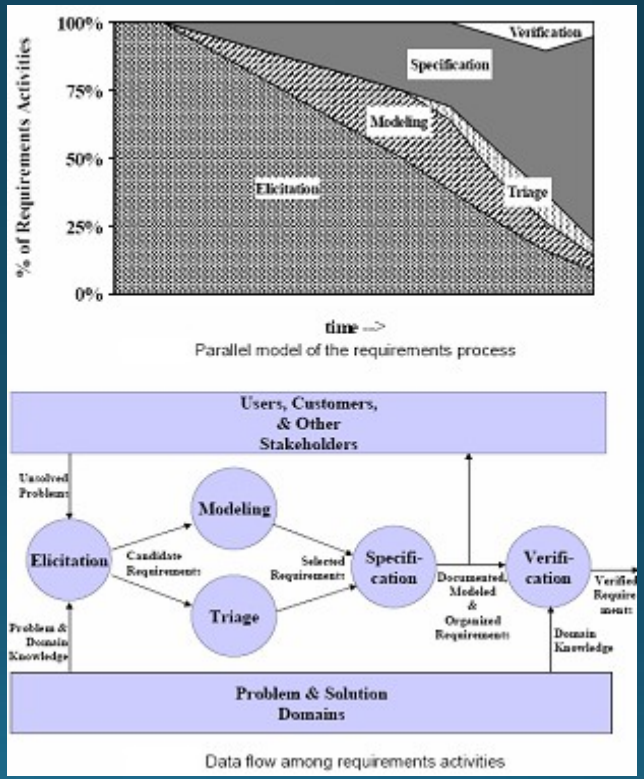
- Requirements elicitation
  - Determining what the customer requires
- Requirements analysis and negotiation
  - Understanding the relationships among various customer requirements and shaping them
- **Requirements specification**
  - Building a tangible model of requirements

Requirements Engineering - 3

- System modeling
  - Building a representation of requirements that can be assessed for correctness, completeness, and consistency
- Requirements validation
  - Reviewing the model
- Requirements management
  - Identify, control and track requirements and the changes that will be made to them

Requirement Activities

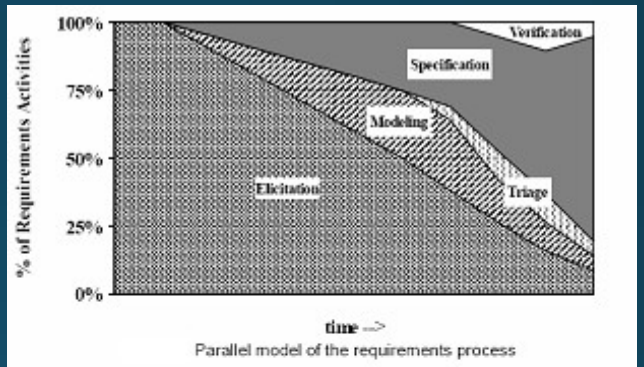
- **Elicitation**: Learning, uncovering, extracting and discovering needs of customers and potential stakeholders
- **Modeling**: Creating & Analyzing models of requirements for increasing understandability and search for incompleteness, correctness, inconsistency, etc.
- **Triage**: Which subset of requirements are appropriate to be addressed in specific releases.



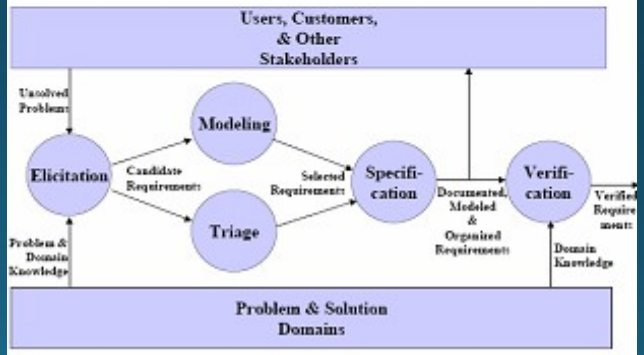
Requirement Activities … cont

- **Specification**: Documentation of desired external behavior of system
- **Verification**: Determining consistency, completeness and other defects in a set of requirements

Requirement activities are not performed sequentially, but in parallel.



Parallel model of the requirements process



Data flow among requirements activities

Specification Levels

- Requirements Definition
  - Natural Language Statements
  - Services expected from System
  - Client Understandable
- Functional Specifications
  - Structured document – sets out the system services in detail
  - Contract between system procurer and developer

Specification Levels

- Software Specifications
  - Abstract description of S/W – basis for design and implementation
  - Use formal specification techniques

Software Requirements Specification (SRS)

- A document containing a complete description of ***what*** the software system will do without describing *how* it will do it.
- May suffer from “what versus how” dilemma.

Levels of Requirements

- Business Requirements
  - Represent high level objectives of the organization or customer requesting the system or product
  - Captured in a document describing the project vision and scope.
- User Requirements
  - Describes tasks the user must be able to accomplish

Levels of Requirements

- Functional Requirements
  - – Define the software functionality the developers must build into the product to enable users to accomplish their tasks - thereby satisfying the business requirements.

Levels of Requirements

- Non-Functional Requirements
  - Include regulations, standards, and contracts to which the product must conform:
    - Description of external interfaces
    - Design and implementation constraints
    - Quality and performance attributes.
  - Constraints are restrictions that are placed on the choices available to the developer for design and construction of the software product.

Levels of Requirements

- All user requirements must align with business requirements
- Requirements do not include design or implementation details.

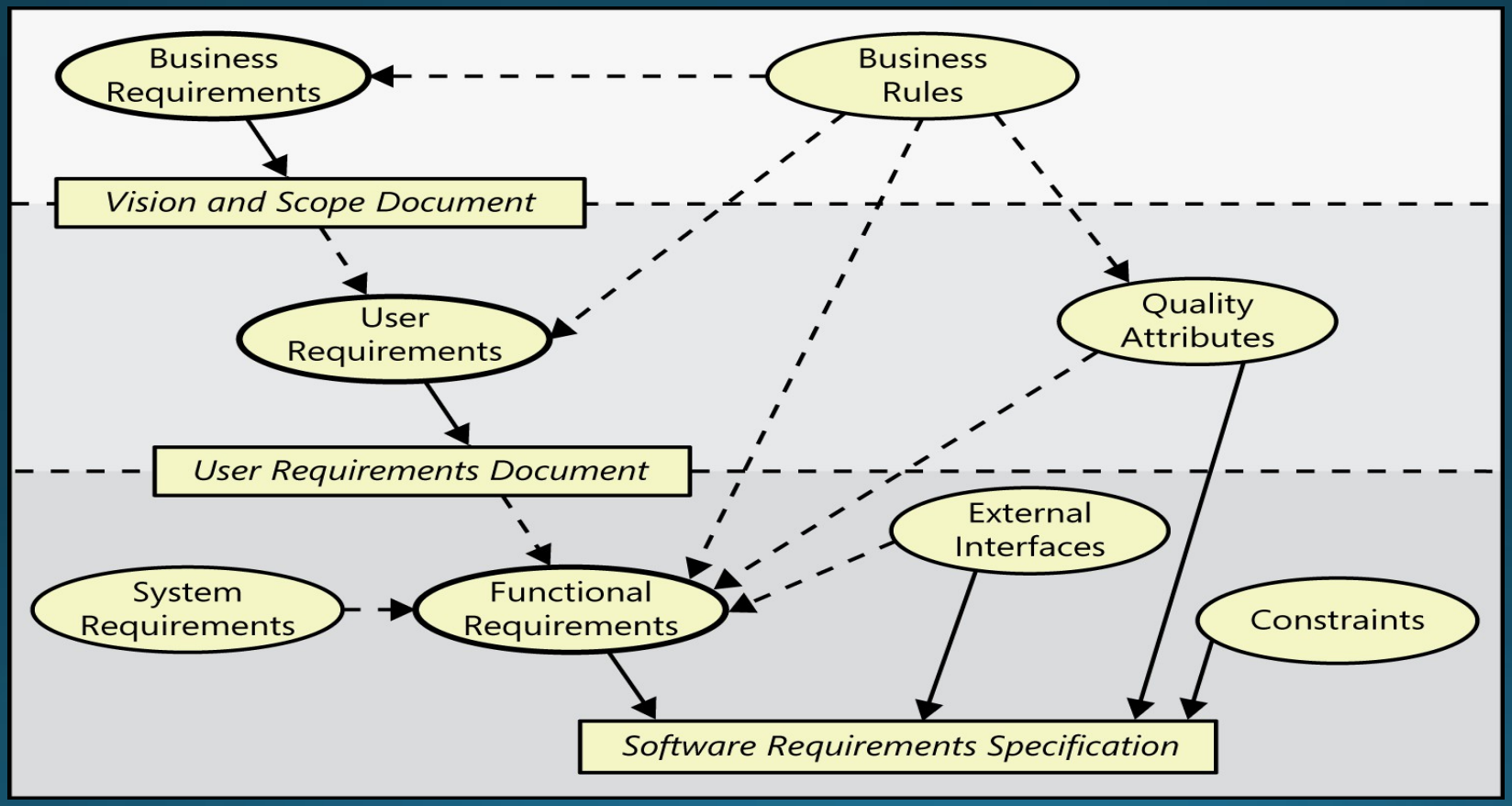
=> Focus on what to build.

Kinds of requirements

(Example)

- BR – user will be able to correct spelling errors in a document efficiently
  - Spell checker is included as a feature
- UR – finding spelling errors in the document and decide whether to replace each misspelled word with the one chosen from a list of suggested words
- FR
  - Find and highlight misspelled words
  - Display a dialog box with suggested replacements
  - Making global replacements
- NFR – It must be integrated into the existing word-processor which runs on windows platform
  - Auto vs. manual correction (user perspective)

Relationships among several types of requirements information



Potential stakeholders within the project team

Outside the Developing Organization

Direct user
Indirect user
Acquirer
Procurement staff
Legal staff
Contractor
Subcontractor
Business management
Contracting officer
Government agency
Subject matter expert
Program manager
Beta tester
General public
Consultant
Compliance auditor
Certifier
Regulatory body
Software supplier
Materials supplier
Venture capitalist

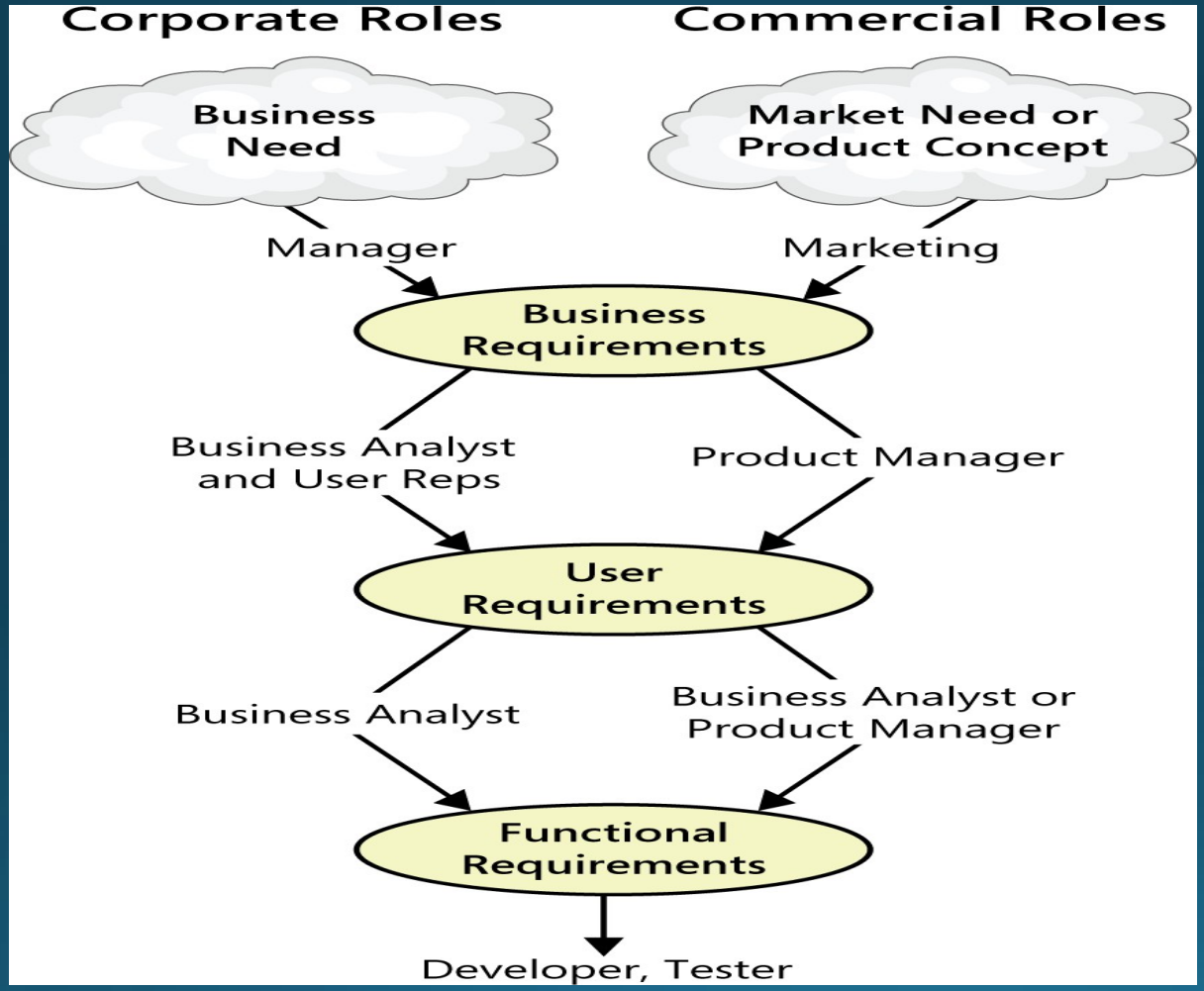
Developing Organization

Development manager
Marketing
Operational support staff
Legal staff
Information architect
Company owner
Sales staff
Installer
Maintainer
Program manager
Usability expert
Subject matter expert
Executive sponsor
Project management office
Manufacturing
Training staff
Portfolio architect
Infrastructure support staff

Project Team

Project manager
Business analyst
Application architect
Designer
Developer
Product owner
Data modeler
Process analyst
Tester
Product manager
Quality assurance staff
Documentation writer
Database administrator
Hardware engineer
Infrastructure analyst
Business solutions architect

Stakeholders participation in Requirements Development



Benefits from 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

Benefits from a high-quality requirements process ..

- Fewer miscommunications
- Reduced scope creep
- Reduced project chaos
- Higher customer and team member satisfaction
- Products that do what they're supposed to do

Requirements Pulse

- The project's business objectives, vision, and scope were never clearly defined.
- Customers were too busy to spend time working with analysts or developers on the requirements.
- Your team could not interact directly with representative users to understand their needs.
- Customers claimed that all requirements were critical, so they didn't prioritize them.
- Developers encountered ambiguities and missing information when coding, so they had to guess.
- Communications between developers and stakeholders focused on user interface displays or features, not on what users needed to accomplish with the software.

Requirements pulse …

- Your customers never approved the requirements.
- Your customers approved the requirements for a release or iteration and then changed them continually.
- The project scope increased as requirements changes were accepted, but the schedule slipped because no additional resources were provided and no functionality was removed.
- Requested requirements changes got lost; no one knew the status of a particular change request.
- Customers requested certain functionality and developers built it, but no one ever uses it.
- At the end of the project, the specification was satisfied but the customer or the business objectives were not.

References

- Software Requirements by Karl Wiegers, 2013
- Requirements Engg Processes and Techniques by Kotonyo & Sommerville