

Software Engineering

CE2006/CZ2006

Requirements Elicitation 1 of 3

Discussion Topics

- Requirements Elicitation
 - Functional & Non-Functional Requirements
 - Data Dictionary
- Requirements Validation
 - Prototype

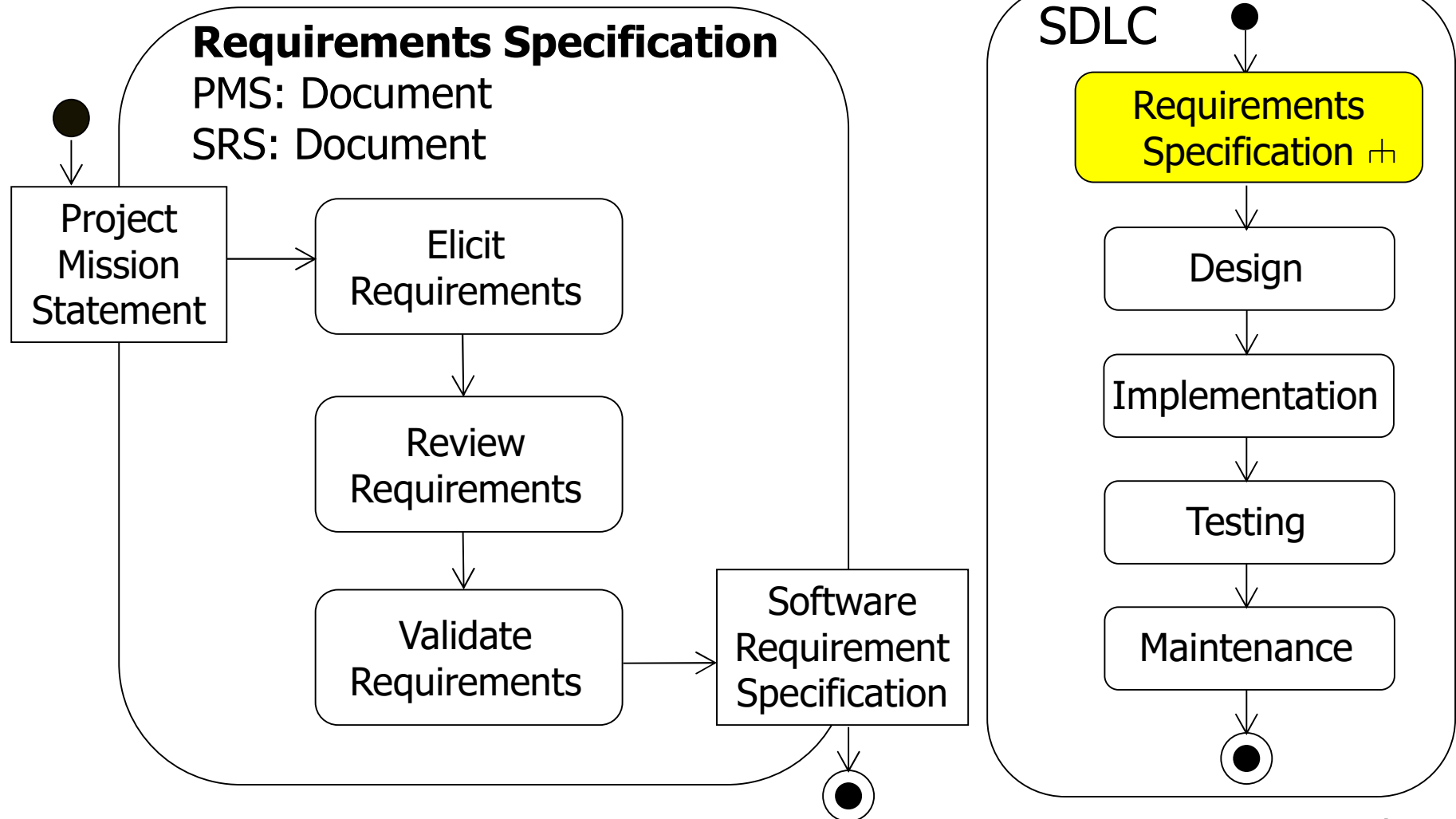
Reading

- Bruegge Chapter 4 sections 4.1-4.3

Correct Requirements Elicitation and Specification are Necessary for a Successful Project



Requirements Specification Activities



Project Mission Statement

A **simple short statement** of **what you intend to accomplish** in your project.

Companies and organisations have mission statements:

Google - to organize the world's information and make it universally accessible and useful.

BMW - the BMW Group is the world's leading provider of premium products and premium services for individual mobility.

Apple – Apple designs Macs, the best personal computers in the world, along with OS X, iLife, iWork and professional software. Apple leads the digital music revolution with its iPods and iTunes online store. Apple has reinvented the mobile phone with its revolutionary iPhone and App store, and is defining the future of mobile media and computing devices with iPad.

Project Mission Statement

A project should have a project mission statement that describes the project in **two or three** sentences.

Typically these sentences define

1. The **problem**: what will be done (scope and limits)
2. The **stakeholders, developers and users**
3. The **outcomes** and **benefits** of the project

Example:

The GoFast team will develop a website that enables airline travellers to rate their travel experiences. This project will be considered complete when the website has been tested and approved for release by the FactFinding Organisation. This project supports the International Travel Watchdogs objective to ensure air passengers can openly compare airlines.

Requirements Elicitation

Eliciting stakeholder needs and desires through:

- Interview
- Observation
- Workshop
- Legacy Product Study
- Competitive Product Study
- Prototype

Learn problem
domain
Study user
tasks

Types of Requirements

- **Functional requirements** describe **interactions** between the system and the environment, to **map program inputs to program outputs**. Basically the things that the system must do.
- **Non-Functional requirements** describe the **properties** the system must have, that is not directly related to the functional behaviour of the system.

Examples of Functional Requirements

System functionality to be performed

e.g., The library member must be able to search the library catalog.

e.g., The bank customer must be able to withdraw cash from the ATM.

Information to be processed

e.g. The system must display the current time in 24 hour format.

e.g. The system must display the temperature in degrees centigrade in the range -10C to +130C to one decimal place of accuracy.

Interface with other systems

e.g. The system must be able to use wifi to communicate all transactions with a clients secure database.

e.g. The system must be able to control up to six robot arms simultaneously.

Examples of Non-Functional Requirements

Usability	Help messages must be displayed in the local language according to the user's locale.
Reliability	After a system reboot, the full system functionality must be restored within 5 minutes.
Performance	When a book is placed in the checkout pad, the system must detect it within 2 seconds.
Supportability	The database must be replaceable with any commercial product supporting standard SQL queries.

Documenting the Requirements

- Use good technical writing style
 - Write **complete, simple, precise, unambiguous** sentences using **active** voice
 - Define terms clearly and use them **consistently**
 - Use **clear layout and formatting** (e.g., organizing the requirements in a hierarchy)
- State requirements in an ***atomic*** manner, such that the specification is ***verifiable*** and ***traceable*** and ***unambiguous***.

Atomic Requirements

When a computer is added, the tracking system requires the user to specify its type and allow the user to provide a description. Both these fields must be text of length >0 and <512 characters.



1.1 When a computer is added to the tracking system, the user must enter the computer type.



1.1.1 The computer type data must be text of at least one character and less than 512 characters.

1.2 When a computer is added to the tracking system, the user must enter a description of the computer.




1.2.1 The description of the computer must be text of at least one character and less than 512 characters.

Requirements Verifiability

Verifiable - sufficiently specific to be testable

The user interface must be user-friendly. 

80% of first-time users must be able to enter a simple search query within 2 minutes of starting to use the system. 

The system must control more than one drill press. 

The system must control up to seven drill presses concurrently. 

Requirements Traceability

- The ability to track requirements from their expression in an SRS (software requirements specification) to their realization in design, documentation, source code and their verification in reviews and tests
- The ability to track dependencies among requirements, system functions and system components

Requirements Validation

Stakeholders – the system specified meets their needs and desires (i.e. correct) and requirements are prioritized

Development Team – requirements (and underlying assumptions) are properly understood

- Review
 - Walkthrough, inspection, critical review
 - Checklist for completeness, consistency, unambiguity, correctness
- Prototype

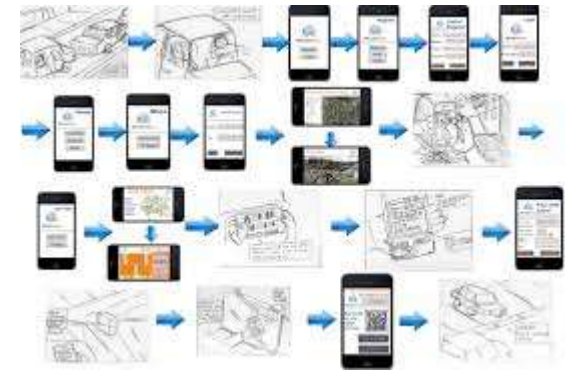
User Interface Prototype

A picture is worth a thousand words.

Sketches or a model of what a system will look like brings the requirements to life for all stakeholders.

Can be implemented using

- Story Boarding – sequence of graphics showing different views of the interface in a specific interaction.



- Still images created on a computer.



- Interactive prototype that illustrates some simulated dialogue

The Data Dictionary

The **problem domain** glossary

- Ensures consistent unambiguous terminology that all stakeholders can agree on

E.g. within a university/college define the terms

- **Programme of Study**
 - **Course**
 - **Degree Programme**
-
- Ensures specialised terms are defined E.g.
 - **Lecture**
 - **Tutorial**
 - **Class**
 - **Laboratory**
 - **Seminar**

Example of Data Dictionary

Term	Definition
Program of Study	A university program that a student enrolls into. There are three levels of programs: undergraduate, master, and PhD.
Semester	There are four semesters in an academic year. Semester 1 and Semester 4 have 13 teaching weeks. Semester 2 and Semester 3 have 5 teaching weeks.,
Course	A course is a basic unit of teaching. A course must be either compulsory or elective. A course must include lectures and tutorials . Some courses may have labs.
Lecture	The traditional form of class that are delivered in lecture theaters. Each lecture will be video recorded and published in the learning management system
Tutorial	The interactive, small-size classes that are conducted in technology-enhanced tutorial rooms. Tutorials will not be video recorded.
....	

Focus on **problem domain** terms,
not implementation terms.