Requirements Engineering

Course Summary
## Course outline

<table>
<thead>
<tr>
<th>No</th>
<th>Date</th>
<th>Topic</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2. September</td>
<td><strong>No lecture</strong></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>9. September</td>
<td>RE framework</td>
<td>Elicitation</td>
</tr>
<tr>
<td>3</td>
<td>16. September</td>
<td>Specification</td>
<td>Negotiation</td>
</tr>
<tr>
<td>4</td>
<td>23. September</td>
<td></td>
<td><strong>Test 1</strong></td>
</tr>
<tr>
<td>5</td>
<td>30. September</td>
<td><strong>No lecture</strong></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>7. October</td>
<td>Management</td>
<td>(Prioritisation, Traceability and Change management)</td>
</tr>
<tr>
<td>7</td>
<td>14. October</td>
<td>Validation</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>21. October</td>
<td></td>
<td><strong>Test 2</strong></td>
</tr>
<tr>
<td>9</td>
<td>28. October</td>
<td>Non-functional reqs</td>
<td>Goal modelling</td>
</tr>
<tr>
<td>10</td>
<td>4. November</td>
<td><strong>No lecture or goal modelling exercise</strong></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>11. November</td>
<td>Scenario modelling</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>18. November</td>
<td></td>
<td><strong>Test 3</strong></td>
</tr>
<tr>
<td>13</td>
<td>25. November</td>
<td>Requirements modelling</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>2. December</td>
<td></td>
<td><strong>Test 4</strong></td>
</tr>
<tr>
<td>15</td>
<td>9. December</td>
<td>Course summary</td>
<td>Introduction to workshop</td>
</tr>
<tr>
<td>16</td>
<td>16. December</td>
<td>Requirements Engineering Workshop</td>
<td></td>
</tr>
</tbody>
</table>
Where are the challenges?

Application Domain

specification

Machine Domain

programs

computers
Elicitation Techniques

- **Traditional techniques**
  - Reading existing documents
  - Analyzing hard data
  - Interviews
    - Open-ended
    - Structured
  - Surveys / Questionnaires
  - Meetings

- **Collaborative techniques**
  - Focus Groups
    - Brainstorming
    - JAD/RAD workshops
  - Prototyping
  - Participatory Design

- **Contextual (social) approaches**
  - Ethnographic techniques
    - Participant Observation
    - Enthnmethodology
  - Discourse Analysis
    - Conversation Analysis
    - Speech Act Analysis
  - Sociotechnical Methods
    - Soft Systems Analysis

- **Cognitive techniques**
  - Task analysis
  - Protocol analysis
  - Knowledge Acquisition Techniques
    - Card Sorting
    - Laddering
    - Repertory Grids
    - Proximity Scaling Techniques
# Specification / Documentation

1. **Introduction**
   - Purpose
   - Scope
   - Definitions, acronyms, abbreviations
   - Reference documents
   - Overview

2. **Overall Description**
   - Product perspective
   - Product functions
   - User characteristics
   - Constraints
   - Assumptions and Dependencies

3. **Specific Requirements**

Appendices

Index

---

**Identifies the product, & application domain**

**Describes contents and structure of the remainder of the SRS**

**Describes all external interfaces: system, user, hardware, software; also operations and site adaptation, and hardware constraints**

**Summary of major functions, e.g. use cases**

**Anything that will limit the developer’s options (e.g. regulations, reliability, criticality, hardware limitations, parallelism, etc)**

**All the requirements go in here (i.e. this is the body of the document). IEEE STD provides 8 different templates for this section**
Requirements negotiation

• **The goal of negotiation**
  – Identify conflicts
  – Analyse the causes of each conflict
  – Resolve the conflicts by means of appropriate strategies
  – Document the conflict resolution and the rationale

• **Resolve conflicts at the goal level**
  – Goals document rationale of the solution-oriented requirements
  – Fundamental contradictions can be resolved before the stakeholders go into technical details (of the goals realisation)
Requirements prioritisation

- **Calculate return on investment**
  - Assess each requirement’s importance to the project as a whole
  - Assess the relative cost of each requirement
  - Compute the cost-value trade-off:

```
<table>
<thead>
<tr>
<th>Priority</th>
<th>Cost (percent)</th>
<th>Value (percent)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Medium</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>High</td>
<td>15</td>
<td>15</td>
</tr>
</tbody>
</table>
```

![Graph showing cost-value trade-off for different priorities](image-url)
Requirements Traceability
Change Management

- **Product (artefact) dimension**
  - Concrete goals, scenarios, and solution-oriented requirements

- **Version dimension**
  - Manages different change states of the artefact of the product dimension
Validation Goal

- Check whether the **outputs** of activities fulfill defined quality criteria
- Check whether the **execution of activities** adheres to process definitions and activity guidelines
- Check whether the **inputs** of activities fulfill defined quality criteria
NFR

• We have to turn our vague ideas about quality into measurables

The Quality Concepts
(abstract notions of quality properties)

Measurable Quantities
(define some metrics)

Counts taken from Design Representations
(realization of the metrics)

examples...

reliability
mean time to failure?
run it and count crashes per hour???

complexity
information flow between modules?
count procedure calls???

usability
time taken to learn how to use?
minutes taken for some user task???
Goal Modelling

• **Relationships between goals:**
  • One goal **helps** achieve another (+)
  • One goal **hurts** achievement of another (-)
  • One goal **makes** another (++)
    • Achievement of goal A guarantees achievement of goal B
  • One goal **breaks** another (---)
    • Achievement of goal A prevents achievement of goal B

• **Goal Elaboration:**
  • “**Why**” questions explore higher goals (context)
  • “**How**” questions explore lower goals (operations)
  • “**How else**” questions explore alternatives
Scenario modelling

| Use Case ID: |  |
| Use Case Name: |  |
| Created By: | Last Updated: |
| Date Created: | Date Last Updated: |

- Actors:
- Description:
- Trigger:
- Preconditions: 1.
- Postconditions: 1.
- Normal Flow: 1.
- Alternative Flows:
- Exceptions:
- Includes:
- Priority:
- Frequency of Use:
- Business Rules:
- Special Requirements:
- Assumptions:
- Notes and Issues:

- Add new staff member
- Add new staff grade
- Change rate for the client
- Change grade for the staff member
- Calculate staff bonuses
Requirements Modelling

Entity-relationship diagram

Data model

Behavioural model

Transition diagram

Functional model

Requirement (natural language)

Entity: entrance door

Entity: glass break detector

State: alarm state

Event: inform security company

Event: entrance door damaged

Data flow diagram

Function: inform security company

If a glass break detector attached to the entrance door detects that the entrance door has been damaged, the system shall enter the alarm state and inform the security company.
Modalities and Assessment

- **Tests** – 10 points each
- **Workshops** – 20 points
- **Exam** – 50 points

To be admitted to the exam, at least 35 points of the grade from the tests and workshops need to be collected during the semester.
Consultation before exam

• (not obligatory!!)

3. January, 14:00-15:00