Requirements Engineering

- Introduction
- Stakeholders, goals and scenarios
- Requirements elicitation
- Requirements modelling
- Requirements specification
- Non-functional requirements
- Requirements prioritisation and risk analysis
- Requirements validation and verification
Definition of RE

Requirements Engineering (RE) is a set of activities concerned with identifying and communicating the purpose of a software-intensive system, and the contexts in which it will be used. Hence, RE acts as the bridge between the real world needs of users, customers, and other constituencies affected by a software system, and the capabilities and opportunities afforded by software-intensive technologies.

- Not a phase or stage!
- Communication is as important as the analysis
- Quality means fitness-for-purpose. Cannot say anything about quality unless you understand the purpose
- Designers need to know how and where the system will be used
- Requirements are partly about what is needed...
- ...and partly about what is possible
- Need to identify all the stakeholders - not just the customer and user

Quality means fitness-for-purpose. Cannot say anything about quality unless you understand the purpose.
... initiate and influence the definition of...

... classify...

... illustrate satisfaction...

... lead to the identification of new...

... lead to revision of...
Requirements Elicitation

1. The project requirements are forming in my mind.
2. Now they're changing... changing... changing... okay. No, wait... changing... changing... done.
3. Naturally, I won't be sharing any of these thoughts with engineering.
4. I budgeted for some goons to beat it out of you.
What is Modelling?
IEEE Standard for Requirements Specification

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
Making Requirements Measurable

• 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)

equalities...

- 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???
AHP example - estimating costs

<table>
<thead>
<tr>
<th></th>
<th>Req1</th>
<th>Req2</th>
<th>Req3</th>
<th>Req4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Req1</td>
<td>1</td>
<td>1/3</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>Req2</td>
<td>3</td>
<td>1</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>Req3</td>
<td>1/2</td>
<td>1/5</td>
<td>1</td>
<td>1/3</td>
</tr>
<tr>
<td>Req4</td>
<td>1/4</td>
<td>1/3</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>

Normalise columns

Result

- Req1 - 26% of the cost
- Req2 - 50% of the cost
- Req3 - 9% of the cost
- Req4 - 16% of the cost

Sum the rows

<table>
<thead>
<tr>
<th>sum</th>
<th>sum/4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.05</td>
<td>0.26</td>
</tr>
<tr>
<td>1.98</td>
<td>0.50</td>
</tr>
<tr>
<td>0.34</td>
<td>0.09</td>
</tr>
<tr>
<td>0.62</td>
<td>0.16</td>
</tr>
</tbody>
</table>

Sum the rows

<table>
<thead>
<tr>
<th>sum</th>
<th>sum/4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.05</td>
<td>0.26</td>
</tr>
<tr>
<td>1.98</td>
<td>0.50</td>
</tr>
<tr>
<td>0.34</td>
<td>0.09</td>
</tr>
<tr>
<td>0.62</td>
<td>0.16</td>
</tr>
</tbody>
</table>
Verification and Validation

- **Validation:**
  - “Are we building the right system?”
  - Does our problem statement accurately capture the real problem?
  - Did we account for the needs of all the stakeholders?

- **Verification:**
  - “Are we building the system right?”
  - Does our design meet the spec?
  - Does our implementation meet the spec?
  - Does the delivered system do what we said it would do?
  - Are our requirements models consistent with one another?