CISC 322
Software Architecture

Lecture 03:
Non Functional Requirements
Emad Shihab
Adapted from: Ahmed E. Hassan
Software Requirements

“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 and Sawyer 1997, Karl E. Wiegers 1999)

Not how the product will be constructed
Software Requirements

- Involve the elicitation, analysis, specification and validation of the different stakeholders

- An analyst/requirement engineer:
  - Elicits these demands/needs (raw requirements)
  - Analyzes them for consistency, feasibility, and completeness
  - Formulates them as requirements and write down a specification
  - Validates that the gathered requirements reflect the needs/demands of stakeholders
Many Stakeholders
Different Visions, Conflicting Goals

Need to be precise, reduce ambiguity
Fixing a Bug During Maintenance

1. Tracking the user
   2. The user no longer in company
   3. The user does not recall rationale

1. Developers may no longer be part of the team
   2. Change may not fit in current arch/design

1. Redistribute
   2. Reinstall
   3. Retrain

1. Retesting
Questions that Arise During Requirement Gathering

- Is this a need or a requirement?
- Is this a nice-to-have vs. must-have?
- Is this the goal of the system or a contractual requirement?
- Do we have to program in Java? Why?
A Good Understanding of the Problem is Essential
A Good Understanding of Problem is Essential

- Elevators in skyscraper
- Toothpaste boxes
- Out of coverage simulator
- Ice cream store in Lake Como (Handicap service)
Software Specification

- Specification acts as a bridge between the real-world environment (demands of stakeholders) and the software system.
Contents of Requirement Specification

Data requirements:
- System state: Database, comm. states
- Input/output formats

Functional requirements, each interface:
- Record, compute, transform, transmit
  Theory: $F(input, state) \rightarrow (output, state)$
- Function list, pseudo-code, activity diagram
- Screen prototype, support tasks xx to yy

Quality reqs:
- Performance
- Usability
- Maintainability
  . . .

Other deliverables:
- Documentation
- Install, convert, train . . .

Managerial reqs:
- Delivery time
- Legal
- Development process . . .

Helping the reader:
- Business goals
- Definitions
- Diagrams . . .
System Perspective Diagram

System perspective is a block diagram that describes the boundaries of the system, its users, and other interfaces.
## Example Constraints

<table>
<thead>
<tr>
<th>Source</th>
<th>Constraint</th>
<th>Rationale</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operational</td>
<td>An exact copy of sales order data must remain on the legacy database for up to one year.</td>
<td>The risk of data loss is too great; we will need to run in parallel for up to one year.</td>
</tr>
<tr>
<td>Systems and OS</td>
<td>The applications footprint on the server must be less than 20 megabytes.</td>
<td>We have limited server memory available.</td>
</tr>
<tr>
<td>Equipment budget</td>
<td>The system must be developed on existing server and host; new client hardware for users may be provided.</td>
<td>Cost control and maintenance of existing systems.</td>
</tr>
<tr>
<td>Personnel budget</td>
<td>Fixed staffing resource; no outsourcing.</td>
<td>Fixed operating costs as per the current budget.</td>
</tr>
<tr>
<td>Technology mandate</td>
<td>New OO methodology to be used.</td>
<td>We believe that this technology will increase productivity and increase reliability of the software.</td>
</tr>
</tbody>
</table>
Example SRS

- Read and fill in groups
- Discussion
Classic: A good requirement spec is:

Correct
Each requirement reflects a need.

Complete
All necessary requirements included.

Unambiguous
All parties agree on meaning.

Consistent
All parts match, e.g. E/R and event list.

Ranked for importance and stability
Priority and expected changes per requirement.

Modifiable
Easy to change, maintaining consistency.

Verifiable
Possible to see whether requirement is met.

Traceable
To goals/purposes, to design/code.

Necessary AND Feasible

Additional:
Traceable from goals to requirements.
Understandable by customer and developer.
Next Class

- Friday, Sep 16
- Will cover:
  - Non-functional requirements