If software is simply for automation, what would a washing machine be like?
Requirements Analysis, Modeling and Specification

- Requirements Analysis, Modeling and Specification
- Problem
- Carving the Solution Space
- Prioritizing Requirements
What is a *Model*?
Requirements Analysis, Modeling & Specification

A continuous loop

Problem Elicitation

Problem Analysis

Diagrams, Charts, Tables

Problem Specification

Text, Notes, Knowledge

Modelling

(often informal models)

(Concrete models)

(reqs. spec. models)

mediating representations
A problem can be defined as the difference between things as *perceived* and things as *desired*.

Understand:
- The perceived
- The problem – *do not* create any unreal problems
- The desired

The perceived + The problem = The desired?
Exercise: Student Application Processing System

perceived (AS-IS)  (Problem)  desired (TO-BE)
A Problem Analysis Roadmap

Enterprise/Business Problem defined

- Identify stakeholders for problem.
- Root cause analysis.

Actual problem identified and defined

- Understand the problem in the context of the business goals.
- Identify constraints on the system/project.

Problem validated/adjusted

- Gain agreement on the problem def. wrt. root causes.
- Consider alternatives & choose the best solution(s) to meet the goals.
- Define the solution system boundary.

Best solution identified

- Reassess that the solution idea is the best solution.

Elicit Requirements

- Expand stakeholder list for solution.

Establish common vocabulary => Glossary with a Domain Model
Carving the Product Space

Requirements represent a compromise.

- User Needs
- Wants
- Developer Perspective (acceptable)
- (unacceptable)
- Environmental
- Economic
- Political
- Feasibility
- Technology Risks (acceptable)
- (unacceptable)
- (unacceptable)
- (acceptable)
- Laws & Standards

Admissible Product Space Solution
Carving the Product Space

Requirements represent a compromise.

Example

E.g., Wired/Wireless phone & PCS

User Needs
- Ordinary
  - Wants: all in one device (phone, pager, TV, computer)
  - Needs: cellular + wired service
- Office worker
  - Wants: video phone + pager + teleconferencing
  - Needs: cellular + wired service + ID

Wants
- Sprint: 2 yr, $2 billion, across US
  - Guarantee w. CDMA only
  - Needs: 3yr, $2B, MANs, CDMA + FDMA

Customer Needs
- Preferred
  - Nortel: Unacceptable
    - Hand-over of ownership
    - Creeping reqs. rate > 1 %
    - Acceptable: nice warranty adaptive (10 new features/yr)

Developer Perspective
- (acceptable)
- (unacceptable)

Technology Risks
- Unacceptable: 95% guarantee w. CDMA only
  - Only 7-layer
- Acceptable: 90% guarantee w. CDMA + FDMA

Laws & Standards
- Unacceptable: towers/cells every 100 meters
  - <-> Zoning law
- Acceptable: use any frequency --> hearing aid
- Acceptable: close to hexagonal, but special BSs in subway, dense areas

Varying degrees of acceptable/unacceptable solutions
What Is the Problem Behind the Problem?

Fishbone Diagram Techniques

- Want Privacy when banking
- Banking in airports
- Want more banking locations
- Too much waiting
- No Banking at night
- Queues in the branches are too long

Customers are dissatisfied with our service.

List contributing causes to the identified problem. Keep asking “Why?” (expand each rib).
We need ATMs. Why?

The perceived problem:

1. Branches are too long
2. Queues in the bank
3. Too much waiting
4. Dissatisfied with our service
5. Customers are dissatisfied
6. Want more banking
7. Want banking in airports
8. No banking at night

What is the problem of your project? Why is your solution the right solution?

List the reasons why the solution is the right solution.

Keep asking "Why?" (expand each rib).
Focus on Largest Contributors - *Pareto’s Law*

20% of the effort yields 80% of the benefit.

Rank in order. Use the 80-20 Rule to focus on the top contributing causes to address the greatest portion of the problem.

*What are in the 20% of the problem of your project?*
D, S achieves R to solve P in D

What next?
**D, S achieves R to solve P in D**

Define Boundaries for the Enterprise/Business and the Solution

- Users
- Maintenance
- Reports

More on this in Enterprise Modeling

Consider Application Processing
D, S achieves R to satisfy/satisfice G (solve P) in D

\[ M^0, \text{Prog}^a \models S^0; S^0, D^0 \models R^0; R^0, D^0 \models G; (G \models \neg P) \lor (G \models \neg P) \]

**Exercise**

**An Application Processing System**

D: include a functional model, a workflow model, an informational model, a BM

P: include complaints (both external and internal), weaknesses, etc.

G: include wants and needs countering P – both hard and soft

R: include an interaction model between D and S

S: include a functional model, an informational model, a behavioral model

**Requirements should contain nothing but information about the environment.**

© Lawrence Chung
Modeling is Everywhere

Problem Elicitation
- exploratory, brain-storming, open-ended thinking
- elaboration of unclear goals and needs
- identification of sources, views, needs & wants

Wicked Problem
- process of understanding real-world problems, how they relate to stakeholder needs, and proposing solutions to meet those needs.

Problem Analysis

Detect Defects → Resolve Defects

Intermediate representations

- determination of "real" users (i.e., identification of sources)
  - (e.g., stratification during sampling, questionnaires & interviewing)
  - customers, tellers, other employees
- clarification of goals (e.g., Goal-directed approach)
- detecting differences in views and integrating them, & recording rationale
- resolving a mismatch of wants and needs
- prioritization of defects
- understanding (all constraints on the) solutions and evaluating them
- risk analysis (e.g., scenarios)

Problem Specification
- choose formal notations
- create a formal model of the requirements

Modeling is NOT Perfect
[adapted from Jackson, 1995, p124-5]
- There will always be phenomena in the model that are not present in the application domain
- There will always be phenomena in the application domain that are not in the model
- Perfecting the model is not always a good use of your time
<table>
<thead>
<tr>
<th>Stakeholder Representative</th>
<th>User</th>
<th>Description</th>
<th>Responsibilities</th>
<th>Success Criteria</th>
<th>Involvement</th>
<th>Deliverables</th>
<th>Comments/ Concerns</th>
</tr>
</thead>
<tbody>
<tr>
<td>Registrar</td>
<td>Kelly Hansen</td>
<td>The Registrar is typically a college-educated professional with full computer skills. The Registrar is trained and experienced with the use of the current batch-oriented registration. The Registrar is responsible for administering course registration for each school term. This includes supervising administrative and data entry personnel. The registrar's primary responsibility will be maintaining student and professor database entries and keeping course registration active. The registrar's office will also be required to perform......</td>
<td>Management reviewer – especially related to functionality and usability of features required by the Registrar staff.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Carving the Product Space

- Needs
- Features
- Software Requirements
- Test Procedures
- Design
- User Doc

An externally observable service by which the system directly fulfills one or more stakeholder requests. It specifies, from a black-box perspective, how the solution interacts with the outside world. (Solution Space)

The system to be built (Problem Space)

Traceability

Problem

Lawrence Chung

©Lawrence Chung

Carving the Product Space

The system to be built

Features

Needs

Software Requirements

Test Procedures

Design

User Doc

An externally observable service by which the system directly fulfills one or more stakeholder requests. It specifies, from a black-box perspective, how the solution interacts with the outside world.

Problem Space

Solution Space
Carving the Product Space

Requirements exist at many levels of abstraction, possibly with diff. terminology

What How

Stakeholder Needs

Product or System Features

Why?

What How

Software Requirements

Why?

What How

Design Spec
Test Procedures
Documentation Plans

One Man’s Ceiling Is Another Man’s Floor!

©Lawrence Chung
Requirements Prioritization


Given \( n \) requirements,
- Create \( n \times n \) matrix
- Compare each pair
  - entry \((i, j)\) =
    - 1 if \( i \) and \( j \) are of equal value
    - 3 if \( i \) is slightly more preferred than \( j \)
    - 5 if \( i \) is strongly more preferred than \( j \)
    - 7 if \( i \) is very strongly more preferred than \( j \)
    - 9 if \( i \) is extremely more preferred than \( j \)
  - entry \((j, i)\) = 1/entry \((i, j)\)

- Estimate the eigenvalues
  - Calculate the sum of each column
  - Divide each entry by the sum of it’s column
  - Calculate the sum of each row
  - Divide each row sum by \( n \)

This gives a value for each requirement based on estimated percentage of total value of the project.
Requirements Prioritization

An Analytic Hierarchy Process (AHP) Approach

Example

<table>
<thead>
<tr>
<th></th>
<th>r1</th>
<th>r2</th>
<th>r3</th>
<th>r4</th>
</tr>
</thead>
<tbody>
<tr>
<td>r1</td>
<td>1</td>
<td>1/3</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>r2</td>
<td>3</td>
<td>1</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>r3</td>
<td>1/2</td>
<td>1/5</td>
<td>1</td>
<td>1/3</td>
</tr>
<tr>
<td>r4</td>
<td>1/4</td>
<td>1/3</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>

normalize columns

<table>
<thead>
<tr>
<th></th>
<th>r1</th>
<th>r2</th>
<th>r3</th>
<th>r4</th>
</tr>
</thead>
<tbody>
<tr>
<td>r1</td>
<td>0.21</td>
<td>0.18</td>
<td>0.18</td>
<td>0.48</td>
</tr>
<tr>
<td>r2</td>
<td>0.63</td>
<td>0.54</td>
<td>0.45</td>
<td>0.36</td>
</tr>
<tr>
<td>r3</td>
<td>0.11</td>
<td>0.11</td>
<td>0.09</td>
<td>0.04</td>
</tr>
<tr>
<td>r4</td>
<td>0.05</td>
<td>0.18</td>
<td>0.27</td>
<td>0.12</td>
</tr>
</tbody>
</table>

sum the rows

<table>
<thead>
<tr>
<th></th>
<th>sum</th>
<th>Sum/4</th>
</tr>
</thead>
<tbody>
<tr>
<td>r1</td>
<td>1.05</td>
<td>0.26</td>
</tr>
<tr>
<td>r2</td>
<td>1.98</td>
<td>0.50</td>
</tr>
<tr>
<td>r3</td>
<td>0.34</td>
<td>0.09</td>
</tr>
<tr>
<td>r4</td>
<td>0.62</td>
<td>0.16</td>
</tr>
</tbody>
</table>

* Also should compute the consistency index, since the pairwise comparisons may be inconsistent.