RE Process

Lawrence Chung
Department of Computer Science/Software Engineering
The University of Texas at Dallas
RE Process:
What is a Process?

- Given input, transforms it into output
- Consist of a set of activities
- Process descriptions are also “process specifications”!
  - Often produced by Requirements Engineers
  - Should be as complete, consistent, clear, agile, …
RE Process: Why?

Quality of product ↔ Quality of Process

- Garbage in garbage out, so get the right requirements

*** Recall: Error Propagation Cycle
RE Process:
Why?

It is more important to understand the problem than the solution. [Albert Einstein]

If software is simply for automation, what would a washing machine be like?
RE Process:
The Basic RE Evolutionary Process

Old Reality
Old Model
Reverse Engineering
Old Implementation
Change in Reality
Change Definition
New Model
Change Incorporation
Legacy Integration
New Implementation
New Reality
Evolution is inevitable – *traceability* is more than a virtue

- Forward traceability
- Backward traceability
RE Process: A Basic Framework [Loucopoulos]

Many variations and extensions

- 3 fundamental activities:
  - **understand**, (formally) **describe**, attain an **agreement** on, the **problem**

  - **Elicitation**: determine *what’s really needed* (F/NF), *why needed* (**??**), *whom* to talk to (*For, Of, By*)
  - **Specification**: produce a (formal) RS model: translate "vague" into "concrete", etc. make various decisions on what & how
  - **Validation**: assure that the RS model satisfies the users’ needs

- Elicitation: determine what’s really needed (F/NF), why needed (??), whom to talk to (For, Of, By)
- Specification: produce a (formal) RS model: translate "vague" into "concrete", etc. make various decisions on what & how
- Validation: assure that the RS model satisfies the users’ needs

(user reqs)
RE Process:
Spiral Model [KotonyaSummerville98]

How many cycles? When to analyze and negotiate? Risk analysis?

- Requirements elicitation: Requirements discovered through consultation with stakeholders
- Requirements analysis and negotiation: Requirements are analysed and conflicts resolved through negotiation
- Requirements documentation: A requirements document is produced
- Requirements validation: The requirements document is checked for consistency and completeness
An RE Process is dominated by human, social and organisational factors

Would you use this process for your project?

Stakeholders/Actors/Agents

ROLES

- Understand problem: Req. engineer, Domain expert, End-user
- Establish outline requirements: Req. engineer, End-user
- Select prototyping system: Software engineer, Project manager
- Develop prototype: Req. engineer, Software engineer
- Evaluate prototype: End-user, Domain expert, Req. engineer, Software engineer

Actions for prototyping [Kotonya&Sommerville98]
RE Process: A RE Process Maturity Model Based on CMM

5 levels

Level 1 - Initial
Ad-hoc requirements engineering; requirements problems are common

Level 2 - Repeatable
Standardised requirements engineering; fewer requirements problems

Level 3 - Defined
Defined process based on best practice; process improvement in place
# IEEE Standard for SRS

<table>
<thead>
<tr>
<th>1 Introduction</th>
<th>2 Overall Description</th>
<th>3 Specific Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Purpose</td>
<td>Product perspective</td>
<td>Identify the product, &amp; application domain</td>
</tr>
<tr>
<td>Scope</td>
<td></td>
<td>Describes contents and structure of the remainder of the SRS</td>
</tr>
<tr>
<td>Definitions, acronyms, abbreviations</td>
<td>Product functions</td>
<td>Describes all external interfaces: system, user, hardware, software; also operations and site adaptation, and hardware constraints</td>
</tr>
<tr>
<td>Reference documents</td>
<td>User characteristics</td>
<td>Summary of major functions</td>
</tr>
<tr>
<td>Overview</td>
<td>Constraints</td>
<td>Anything that will limit the developer’s options (e.g. regulations, reliability, criticality, hardware limitations, parallelism, etc)</td>
</tr>
<tr>
<td></td>
<td>Assumptions and Dependencies</td>
<td>All the requirements go in here (i.e. this is the body of the document). IEEE STD provides 8 different templates for this section.</td>
</tr>
</tbody>
</table>

**Appendices**

**Index**

3.1 External Interface Requirements

3.1.1 User Interfaces
3.1.2 Hardware Interfaces
3.1.3 Software Interfaces
3.1.4 Communication Interfaces

3.2 Functional Requirements

For example:

3.2.1 Mode 1
   3.2.1.1 Functional Requirement 1.1
   ...

3.2.2 Mode 2
   3.2.1.1 Functional Requirement 1.1
   ...

3.2.n Mode n
   ...

3.3 Performance Requirements

Remember to state this in measurable terms!

3.4 Design Constraints

3.4.1 Standards compliance
3.4.2 Hardware limitations
   etc.

3.5 Software System Attributes

3.5.1 Reliability
3.5.2 Availability
3.5.3 Security
3.5.4 Maintainability
3.5.5 Portability

3.6 Other Requirements
RE in Agile Methods

- Basic Philosophy
  - Reduce communication barriers
    Programmer interacts with customer
  - Reduce document-heavy approach
    Documentation is expensive and of limited use
  - Have faith in the people
    Don’t need fancy process models to tell them what to do!
  - Respond to the customer
    Rather than focussing on the contract

- Weaknesses
  - Relies on programmer’s memory
    Code can be hard to maintain
  - Relies on oral communication
    Mis-interpretation possible
  - Assumes single customer representative
    Multiple viewpoints not possible
  - Only short term planning
    No longer term vision

Example: Extreme Programming
- Instead of a requirements spec, use:
  - User story cards
  - On-site customer representative
- Pair Programming
- Small releases
  - E.g. every three weeks
- Planning game
  - Select and estimate user story cards at the beginning of each release
- Write test cases before code
- The program code is the design doc
  - Can also use CRC cards (Class-Responsibility-Collaboration)
- Continuous Integration
  - Integrate and test several times a day

Would you always use an agile process?
Appendix
RE Processes:
Volere Requirements Process

How many cycles? When to analyze and negotiate?
RE Processes:
RE Process Variability

Many Variety …and Evolution is inevitable

- RE processes vary radically from one organisation to another
- Factors contributing to this variability include:
  - Technical maturity
  - Disciplinary involvement
  - Organisational culture
  - Application domain
  - ...

- There is therefore no ‘ideal’ requirements engineering process [KotonyaSummerville98]
NFRs & RE Process:
A Requirements Management System
Many variations and extensions