Changes in Requirements Engineering

Meghana Dinesh Kumar
UWaterloo ID: 20676763
CS 846
Contents

• Introduction
• Changes during development
• Factors and source of change
• Phases of change management
• Theory into practice
• Examples
• Summary
Requirements Engineering

• Process of defining, documenting and maintaining requirements.

• Analysis involves determining the needs or conditions to meet for a new or altered product or project, taking account of the possibly conflicting requirements of the various stakeholders, analyzing, documenting, validating and managing software or system requirements.
Changes in Requirements Engineering

• Changes always happen at all phases in a project’s life cycle.
• Goal: reduce number of negative influences that generate changes.
• Changes towards the end of a project become uncomfortable. Cost for rectification of mistakes increases.
• Attempts are made to disallow changes to project “after a certain point” which generally coincides with the end of project.
Changes during development

![Graph showing changes during development with 'ideal' and 'critical' lines.](image_url)
Lehman's laws of software evolution

These laws describe a balance between forces driving new developments on one hand, and forces that slow down progress on the other hand.

• **The law of continuing change**: Systems that are used must change, or else automatically become less useful.

• **The law of increasing complexity**: Through changes, the structure of a system becomes ever more complex, and more resources are required to simplify it.
Factors influencing change

- Cost or budget levels
- Resource situation
- Scheduling
- Conceptual changes
- Strategic changes
- External factors, such as new vendors, technologies, or methodologies
- Ambiguous requirements
- Incorrect / Incomplete requirements
- Implementation problems

Creating software requirements is like hiking in a gradually lifting fog. At first only the surroundings within a few feet of the path are visible, but as the fog lifts, more and more of the terrain can be seen.
“Cost was measured in days and defined as the difference between any unused portion of the previous estimate (if it existed) and the effort required to implement the change. This was agreed between the customer and the software development organization. Expressing value in monetary terms was impossible in most situations.”

~ McGee, Sharon, and Des Greer.
<table>
<thead>
<tr>
<th>Change Domain</th>
<th>Trigger</th>
<th>Uncertainty</th>
</tr>
</thead>
<tbody>
<tr>
<td>Market</td>
<td>• Change to Government Policy or Regulation</td>
<td>• Market Stability</td>
</tr>
<tr>
<td></td>
<td>• Changes to Market Demands</td>
<td>• Differing Customer Needs</td>
</tr>
<tr>
<td></td>
<td>• Response to Competitor</td>
<td></td>
</tr>
<tr>
<td>Customer</td>
<td>• Strategic Change</td>
<td>• Stability of Customer’s Business Environment</td>
</tr>
<tr>
<td>Organisation</td>
<td>• Company Reorganisation</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Change in Political Climate</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Customer Hardware/Software Change</td>
<td></td>
</tr>
<tr>
<td>Project</td>
<td>• Business Process Change</td>
<td>• All Stakeholders Involved</td>
</tr>
<tr>
<td>Vision</td>
<td>• Change to Business Case</td>
<td>• Clarity of Shared Product Vision</td>
</tr>
<tr>
<td></td>
<td>• Cost/Schedule Overrun</td>
<td>• Unknown Customer Project Dependencies</td>
</tr>
<tr>
<td></td>
<td>• New Opportunity</td>
<td>• All Stakeholders Identified</td>
</tr>
<tr>
<td>Requirements</td>
<td>• Change of Stakeholder Representative</td>
<td>• Degree of Change to Customers Workflow</td>
</tr>
<tr>
<td>Specification</td>
<td>• New Stakeholder role</td>
<td>• Project Size</td>
</tr>
<tr>
<td></td>
<td>• Participative learning</td>
<td></td>
</tr>
<tr>
<td>Requirements</td>
<td>• Increased Customer Understanding</td>
<td>• Availability of Communication with Customer</td>
</tr>
<tr>
<td>Specification</td>
<td>• First Engagement of Particular User Representative</td>
<td>• Insufficient Sample of User Representatives</td>
</tr>
<tr>
<td></td>
<td>• Increased Developer Understanding of Problem</td>
<td>• Quality of Communication between Analyst/Customer</td>
</tr>
<tr>
<td></td>
<td>• Resolution of Misunderstanding</td>
<td>• Analysis Techniques</td>
</tr>
<tr>
<td></td>
<td>• Resolution of Mis-Communication</td>
<td>• Development Team Knowledge of Business Area</td>
</tr>
<tr>
<td></td>
<td>• Incorrect Requirement Identified</td>
<td>• Quality of Requirements Specification</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Analyst Skill/Experience</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Development team stability</td>
</tr>
<tr>
<td>Solution</td>
<td>• Understanding Technical Solution</td>
<td>• Technical Uncertainty of Solution</td>
</tr>
<tr>
<td></td>
<td>• New Tools/Technology (component)</td>
<td>• Technical Complexity of Solution</td>
</tr>
<tr>
<td></td>
<td>• Design Improvement/Solution Elegance</td>
<td>• COTS Usage</td>
</tr>
</tbody>
</table>

Software requirements change source
Phases of change management: Informing and approval-based

- **Informing**
  - No contract
  - Frequent changes
  - No change requests
  - No formal decision necessary
  - Project just started

- **Approving**
  - Contract exists
  - Changes less frequent
  - Change requests exist
  - Formal decision necessary
  - Changes have large impact (schedule/cost/quality)
...Phases of change management
Attributes of change request:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Owner</td>
<td></td>
</tr>
<tr>
<td>Result of analysis</td>
<td></td>
</tr>
<tr>
<td>Reference</td>
<td></td>
</tr>
<tr>
<td>Date of decision</td>
<td></td>
</tr>
<tr>
<td>Status</td>
<td></td>
</tr>
<tr>
<td>Rationale</td>
<td></td>
</tr>
<tr>
<td>ID</td>
<td></td>
</tr>
<tr>
<td>Date of registration</td>
<td></td>
</tr>
<tr>
<td>Priority</td>
<td></td>
</tr>
<tr>
<td>Proposer</td>
<td></td>
</tr>
<tr>
<td>Short description</td>
<td></td>
</tr>
<tr>
<td>Explanation</td>
<td></td>
</tr>
</tbody>
</table>
Analysis of change requests

- Aim: evaluate the effects of change request
- Create a basis for deciding whether the changes can also be implemented.
- Performed by project manager or experienced developer.
- Present the effects of change in such a way that the decision makers can understand (not too technical).
Decision to implement

• Decision is conducted at various levels.

• Following factors influence:
  • Results of analysis
  • Technical structure
  • Organizational structure of customer

• If decision is to approve, the changes must be entered in the customer and supplier specifications, and produce a new version of the document.
Theory into practice

• Preventing bureaucracy in change management.
  • avoid “extra work”.

• Non-disclosure of changes.
  • Change is regarded negative, therefore kept secret.

• No explicit transition from informing to approval-based change management.
  • Customers want to keep things open to incorporate changes "free of charge”.

• Due to time constraints, changes just before the end of project are no longer documented.
  • This negative effect materializes in follow-up project.
Consequences of lack of change-management

- Greater the labor, more serious the effect.
- Development incorrectly conceived due to lack of information.
- Example: small change to a module of a larger program could cost €100,000.
- Psychological point of view: customer has to feel informed and involved in the entire decision process.
- Customer acceptance may become difficult.
Example: Software development

• Change request: Users often report bugs or desire new functionality from their software programs.

• Product software company: technical and economical feasibility of implementation.

• Creation and/or alteration of software code: other code fragments change too.

• Software and documentation released.

• Project manager verifies the change.
Example: Car manufacturing domain

- Manufacturing: confronted with many changes due to increasing and worldwide competition, technological advances and demanding customers.

- Example:
  - vehicle’s air bags are found to automatically fill with air after driving long distances.
  - This produces a change request (that will justify a change).
  - We perform cost and benefit analysis.
  - Then approve change request.
Stabilizing requirements

• Joint application design (JAD):
  • user and development representatives work together with a facilitator to produce joint requirements specification.

• Prototypes:
  • Many changes occur after clients and users see an application’s interface and output.
  • Building early prototypes can help move some changes to the front of the development cycle.

• Change-control boards:
  • group of managers, client representatives, and technical personnel who decide which change to accept and which to reject.
  • Change-control boards are often encountered in the military and systems-software domains.
Summary

Change management is the process of requesting, determining attainability, planning, implementing, and evaluating of changes to a system.

System requirements reflect the world outside of the system. As this constantly changes, the requirements inevitably change.
References

• Understanding software maintenance and evolution by analyzing individual changes: A literature review.
• Hinley 1996.
• A Few Reasons Why Project Changes Occur By ExecutiveBrief ITworld | NOVEMBER 18, 2008.