Requirements Management
Best Practices

Sponsored by:

Karl Wiegers
Principal Consultant, Process Impact
www.processimpact.com
Facilitates multiple entry methods for entering Requirements into the tool

Task based workflow facilitates your Requirements creation, review and approval process

Allows extensive collaboration during review and approval steps

Provides End to End Traceability to Design, Test Cases & Defects

Manages change request process
Featured Speaker

Karl Wiegers
Principal Consultant, Process Impact
www.processimpact.com

Phone #: 503-698-7879
E-mail: karl@processimpact.com
Source Book


tinyurl.com/reqs3

For 25% discount on Process Impact Goodies Collection:
2. Add Goodies Collection to cart
3. Enter discount code **GOOD17**
Components of Requirements Engineering

- Requirements Engineering
  - Requirements Development
  - Requirements Management

  - Elicitation → Analysis → Specification → Validation
Key Requirements Management Practices

- Create a requirements baseline.
- Manage versions of requirements documents.
- Adopt and enforce a change control process.
- Perform requirements change impact analysis.
- Store requirement attributes.
- Track the status of each requirement.
- Trace requirements into designs, code, and tests.
- Use a requirements management tool.
The Requirements Baseline

◆ **Baseline**: A reviewed, approved, and agreed-upon set of requirements committed to a specific product release.

◆ “Sign-off” is a matter of approving the baseline.

◆ When a baseline is defined:
  - formal change control begins
  - managers make schedule commitments
  - managers determine the staff and budget needed to meet their schedule commitments
Requirements Version Management

◆ Place requirements documents under version control.
  ✓ keep requirements documentation up to date
  ✓ everyone must have access to current versions
  ✓ restrict document update access to authorized individuals

◆ **Best:** Store requirements in a database.

◆ **Better:** Store documents in a configuration management system.

◆ **Good:** Define a version identification scheme.
  
  #1 = “version 1.0 draft 1”
  #2 = “version 1.0 draft 2”
  #n = “version 1.0 approved”
  #n+1 = “version 1.1 draft 1” or “version 2.0 draft 1”
Requirements Change Control

- Uncontrolled changes cause problems:
  - rework, degraded quality, unpredictable schedules

- Define a requirements change process:
  - propose, review, approve, and incorporate changes
  - define state-transition model for allowed change states
  - include impact analysis
  - support with a tool, but *a Tool Is Not a Process!*

- All change requests must follow the process.

- Requirements changes may require renegotiating project commitments.
Possible Change Request Statuses

1. **Submitted**
2. **Evaluated**
   - **Rejected**
3. **Approved**
4. **Change Made**
   - **Canceled**
5. **Verified**
6. **Closed**
A Change Control System

User

defect report, enhancement, requirement change

e-mail with entry

e-mail with response

change

develop

e-mail with entry

change

Customer Reps

Support Staff

database

report

response
Change Control Board

- **Diverse group**
  - ✓ development
  - ✓ project management
  - ✓ documentation
  - ✓ testing
  - ✓ customer
  - ✓ others?

- **Authorized to make binding decisions**

- **Adopt a CCB Charter**
  - ✓ purpose, scope of authority, membership, meeting frequency, decision-making process, communicating status

- **Consider change requests periodically**
  - ✓ request impact analysis
  - ✓ make and communicate accept/reject decisions
  - ✓ set priorities or targeted releases
Managing Change on Agile Projects

Prioritised Product Backlog

- **Current Iteration**
  - New requirement or task
  - Reprioritised requirement or task

- **Next Iteration**
  - Lower priority requirement or task

- **Future Iterations**
  - New requirement or task
  - Deleted requirement or task
Impact Analysis for Requirements Changes - 1

- Identify conflicts with existing requirements.
- Identify affected design, code, test components.
- Assess impact on user interface, database, reports, files, help screens, publications.
- Identify other systems, libraries, or hardware affected.
- Determine which work products will require reviewing.
- Identify plans to update.
Impact Analysis for Requirements Changes - 2

- Will the change affect performance or other quality attributes?
- Is the change technically feasible?
- Will the change overload computer resources for development, test, or host environment?
- Will you have to discard other completed work?
- Does it violate any business rules?
- Does the change affect any other current tasks?
Impact Analysis for Requirements Changes - 3

- Estimate total labor hours for all tasks to be performed.
  - create new designs, code, tests, UI, database, files, reports
  - modify existing designs, code, tests, UI, database, files, reports
  - develop and evaluate prototype
  - retesting
  - reviews and rework

- Allocate resources to tasks.

- Sequence tasks and identify predecessors.

- Determine if change is on critical path.

- Estimate schedule and cost impact from effort.
Requirement Attributes

- Store additional information about each requirement.
- Some suggestions:
  - status
  - priority
  - date created and version number
  - author and person responsible for the requirement
  - origin or rationale behind the requirement
  - allocated subsystem, release, iteration, or build
  - risk
  - validation method
- Track project status through requirements status.
<table>
<thead>
<tr>
<th>Status</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Proposed</strong></td>
<td>requirement was requested by a legitimate source</td>
</tr>
<tr>
<td><strong>Approved</strong></td>
<td>requirement was analyzed, impact evaluated, and allocated to a baseline</td>
</tr>
<tr>
<td><strong>Implemented</strong></td>
<td>code was designed, written, and tested</td>
</tr>
<tr>
<td><strong>Verified</strong></td>
<td>requirement was shown to be implemented correctly in the product</td>
</tr>
<tr>
<td><strong>Deleted</strong></td>
<td>planned requirement was deleted from the baseline</td>
</tr>
<tr>
<td><strong>Rejected</strong></td>
<td>requirement was requested, not approved</td>
</tr>
</tbody>
</table>
Each requirement must be uniquely identified: 3.1.4.2, FR-117, Print.ConfirmCopies
### Benefits:

1. No requirements are overlooked during design and implementation.
2. You can see at a glance what work has been completed.
3. If a test fails, it points to the code to search for the problem.
4. A requirement change points to the affected design, code, and test elements.

<table>
<thead>
<tr>
<th>Req.</th>
<th>Design Element</th>
<th>Source File</th>
<th>Function</th>
<th>Test Case</th>
</tr>
</thead>
<tbody>
<tr>
<td>FR-117</td>
<td>DFD 8.8.7</td>
<td>progmgr.c</td>
<td>execute_action, select_manage</td>
<td>action.1, action.3</td>
</tr>
</tbody>
</table>
## Requirements Traceability Matrix - 2

<table>
<thead>
<tr>
<th>Functional Requirements</th>
<th>Use Cases</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>UC-1</td>
</tr>
<tr>
<td>FR-1</td>
<td></td>
</tr>
<tr>
<td>FR-2</td>
<td></td>
</tr>
<tr>
<td>FR-3</td>
<td></td>
</tr>
<tr>
<td>FR-4</td>
<td></td>
</tr>
<tr>
<td>FR-5</td>
<td></td>
</tr>
<tr>
<td>FR-6</td>
<td></td>
</tr>
</tbody>
</table>
Limitations of Requirements Documents

- Difficult to handle requirements for multiple releases
- Difficult to move a requirement from one baseline to another
- Difficult to reuse requirements information
- Difficult to link information in multiple locations across documents
- Difficult to give stakeholders access to current versions
Typical RM Tool Capabilities

◆ Manage versions and changes
  ✓ version history of every requirement
  ✓ baselining capability

◆ Store requirements attributes
  ✓ system and user-defined
  ✓ filter to view requirements with specific attribute values

◆ Define traceability links
  ✓ requirements to other requirements, designs, tests, etc.
  ✓ assist with change impact analysis

◆ Control access
  ✓ group and individual permissions
  ✓ web access to requirements database
Getting the Most from Your RM Tool

- Write good requirements first.
- Don’t expect the tool to replace a requirements process.
- Expect a culture change.
- Don’t create too many requirement types or attributes.
- Train the tool users.
- Assign responsibilities.
- Take good advantage of tool features.
Requirements Management Best Practices

NO SURPRISES!
- See Karl’s recommendations live in action

- Review the product [https://www.kovair.com/alm-studio/requirements-management/](https://www.kovair.com/alm-studio/requirements-management/)

- Experience an enterprise class RM capability
Karl Wiegers
www.processimpact.com

For 25% discount on Process Impact Goodies Collection:
1. Go to www.processimpact.com/goodies.shtml
2. Add Goodies Collection to cart
3. Enter discount code GOOD17