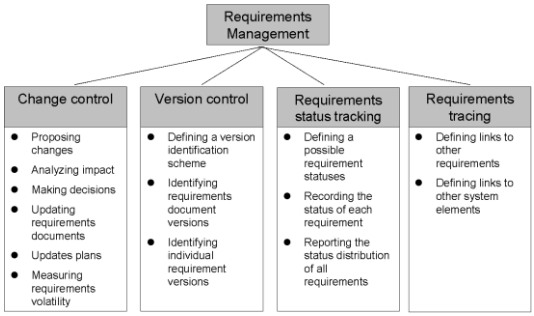
Requirement Management

TOPIC # 11
Chapter 27, 28,29 & 32– Karl Wiegers
Chapter 15,16,17 - Reference

Requirements Management Activities



Requirements version control

- A requirements *baseline* is a set of requirements that stakeholders have agreed to, often defining the contents of a specific planned release or development iteration.
- At the time a set of requirements is baselined—typically following review and approval—the requirements are placed under configuration (or change) management.
- Subsequent changes can be made only through the project's defined change control procedure.
- Version control—uniquely identifying different versions of an item—applies at the level of both individual requirements and requirements sets, most commonly represented in the form of documents.
- Begin version control as soon as you draft a requirement or a document so you can retain a history of changes made.

Tracking Requirement Status

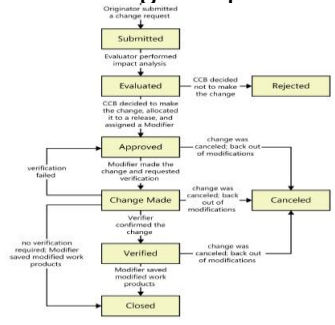
Status	Definition
Proposed	The requirement has been requested by an authorized source.
In Progress	A business analyst is actively working on crafting the requirement.
Drafted	The initial version of the requirement has been written.
Approved	The requirement has been analyzed, its impact on the project has been estimated, and it has been allocated to the baseline for a specific release. The key stakeholders have agreed to incorporate the requirement, and the software development group has committed to implement it.
Implemented	The code that implements the requirement has been designed, written, and unit tested. The requirement has been traced to the pertinent design and code elements. The software that implemented the requirement is now ready for testing, review, or other verification.
Verified	The requirement has satisfied its acceptance criteria, meaning that the correct functioning of the implemented requirement has been confirmed. The requirement has been traced to pertinent tests. It is now considered complete.
Deferred	An approved requirement is now planned for implementation in a later release.
Deleted	An approved requirement has been removed from the baseline. Include an explanation of why and by whom the decision was made to delete it.
Rejected	The requirement was proposed but was never approved and is not planned for implementation in any upcoming release. Include an explanation of why and by whom the decision was made to reject it.

Change Management

Project Roles in change management activities

Role	Description and responsibilities
CCB Chair	Chairperson of the change control board; generally has final decision-making authority if the CCB does not reach agreement; identifies the Evaluator and the Modifier for each change request
CCB	The group that decides to approve or reject proposed changes for a specific project
Evaluator	Person whom the CCB Chair asks to analyze the impact of a proposed change
Modifier	Person who is responsible for making changes in a work product in response to an approved change request
Originator	Person who submits a new change request
Request Receiver	Person who initially receives newly submitted change requests
Verifier	Person who determines whether the change was made correctly

Change Request



Requirements Change Factors

- Requirements errors, conflicts, and inconsistencies
  - May be detected at any phase (when requirements are analyzed, specified, validated, or implemented)
- Evolving customer/user knowledge of the system
  - When the requirements are developed, customers/users simultaneously develop a better understanding of what they really need
- Technical, schedule, or cost problems
  - Difficult to plan and know everything in advance
  - We may have to revisit the list of requirements and adapt it to the current situation

7

Requirements Change Factors

Changing customer priorities, new needs

- May be caused by a change in the system environment (technological, business, political...), i.e., the context
- Business and strategic goals may change
- May be caused by the arrival of a new competitor
- Laws and regulations may change
- Collaborating systems may change
- May also be caused by technology changes in the enterprise (migration to a new operating system, DBMS…)
- May be caused by organizational changes (organizational structure, business processes, employees…)

8

Some Problems Due to Changing Requirements

- Requirements changing towards the end of development without any impact assessment
- Unmatched/outdated requirements specifications causing confusion and unnecessary rework
- Time spent coding, writing test cases or documentation for requirements that no longer exist

9

Version Control

- Another essential aspect of requirements management
  - Every version of a requirement needs to be uniquely identified
  - The last version of a requirement must be available to all team members
  - Changes need to be documented and clearly communicated
  - A version identifier must be updated with every change to the requirement
- Requirements documents should include
  - A revision history: changes, dates, by whom, why...
  - Standard markers for revisions (e.g., strikethrough or underlined text, coloring, line markers…)
- Version control tool may be used
  - To store and manage the revision history
  - To store justifications (to add, modify, delete, reject a requirement)

Traceability

Requirements cannot be managed effectively without requirements traceability

A requirement is traceable if you can discover who suggested the requirement, why the requirement exists, what requirements are related to it, and how that requirement relates to other information such as systems designs, implementations and user documentation

Types of Traceability

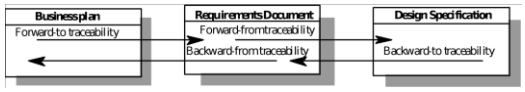
- Requirements – source traceability
  - Links requirements with a person or document
- Requirements – rationale traceability
- Requirements – requirements traceability
  - Links requirements with other requirements which are, in some way, dependent on them
- Requirements – architecture traceability
  - Links requirements with the subsystems where these requirements are implemented (particularly important where subsystems are being developed by different subcontractors)
- Requirements – design traceability
  - Links requirements with specific hardware or software components in the system which are used to implement the requirement

Types of Traceability

- Requirements – interface traceability
  - Links requirements with the interfaces of external systems which are used in the provision of the requirements
- Requirements – feature traceability
- Requirements – tests traceability
  - Links requirements with test cases verifying them (used to verify that the requirement is implemented)
- Requirements – code traceability
  - Generally not directly established, but can be inferred

Backward and Forward Traceability

- Backward traceability
  - To previous stages of development
  - Depends upon each element explicitly referencing its source in earlier documents
- Forward traceability
  - To all documents spawned by a document
  - Depends upon each element in the document having a unique name or reference number



Backward and Forward Traceability

Top to bottom from requirements' point of view

- Forward-to traceability
  - Links other documents (which may have preceded the requirements document) to relevant requirements
  - Help validation
  - Help evaluate which requirements are affected by changes to users' needs
- Forward-from traceability
  - Links requirements to the design and implementation components
  - Help assure that all requirements have been satisfied

Backward and Forward Traceability

Bottom to top from requirements' point of view

- Backward-to traceability
  - Links design and implementation components back to requirements
  - Help determine why each item is designed/implemented
- Backward-from traceability
  - Links requirements to their sources in other documents or people
  - Help validation
  - Help evaluate how changes to requirements impact stakeholders needs

16

Representation – Traceability Table

- Show the relationships between requirements or between requirements and other artifacts
- Table can be set up to show links between several different elements
- Backward and forward traceability
-

User Requirement	Functional Requirement	Design Element	Code Module	Test Case
UC-28	catalog.query.sort	Class Catalog	catalog.sort()	search.7 search.8
UC-29	catalog.query.import	Class Catalog	catalog.import(), cat catalog.validate()	search.12 search.13 search.14

Representation – Traceability Matrix

- Define links between pairs of elements
  - E.g., requirements to requirement, use case to requirement, requirement to test case…
- Can be used to defined relationships between pairs
  - E.g., specifies/is specified by, depends on, is parent of, constrains…
- More amenable to automation than traceability table

Depends-on

	R1	R2	R3	R4	R5	R6
R1			*	*		
R2					*	*
R3				*	*	
R4		*				
R5						*
R6						

Representation – Traceability List

- Traceability matrices become more of a problem when there are hundreds or thousands of requirements as the matrices become large and are sparsely populated
- A simplified form of a traceability matrix may be used where, along with each requirement description, one or more lists of the identifiers of related requirements are maintained

Requirement	Depends-on
R1	R3, R4
R2	R5, R6
R3	R4, R5
R4	R2
R5	R6

Types of Traceability Links

- Note the types of links in the previous examples, as well as the types of objects they relate
  - Satisfies, Tests
  - Refines, References, Contains...
- Others could be created

Requirements	Design	Code (software)	Documentation	Test cases
... 5.1. Braking distance <50 m when speed ~90 km/h 5.2. Absorbers should be electronically controlled.		public RES control(String args[]) throws Exception { Class c = null; if (args.length == 1) {	... Braking: The driver should push brakes sharply to the utmost. ...	Braking test: - on dry asphalt; - on slippery roads - on bumpy roads

Requirement risks

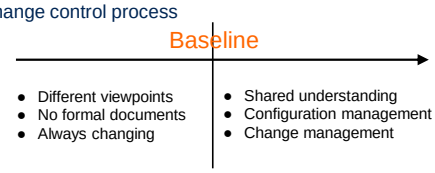
- Including design in specification
- Ambiguous terminology
- Requirement understanding
- Time pressure to proceed despite open issues
- Invalidated requirement
- Inspection proficiency
- Changing requirements. Scope creep
- Requirement change process
- Unimplemented requirements
- Prioritization errors
- Solution presented as needs

Baseline

- Non-modifiable (read-only) version of a document
  - Describes a moment in time
  - May include multiple documents at the same time
- Enables document comparison and management
- Comes with a change history for the document
  - Information on objects, attributes, and links created, deleted, or edited since the creation of the baseline
  - Often also contains information on user sessions (when the document was opened, by whom…)
- Requires access control
- It is advisable to establish a baseline for a new document that is imported into the document management system
  - In order not to lose any changes

Baseline for Requirements

- Represents the set of functional and non-functional requirements that the development team has committed to implement in a specific release
- Before going into the baseline, the requirements should be reviewed and approved by stakeholders
- Once in the baseline, all changes should follow a defined change control process



23

Baseline Usage

- Baselines may be
  - Created
    - Complete image of requirements state at a given time
  - Deleted
  - Visualized
    - Possibility to go back
  - Compared
    - To see changes since a certain time
  - Copied
  - Signed
    - For authorization, contract

24

Change Management

Change Management

- Concerned with the procedures, processes, and standards which are used to manage changes to a system requirements
- Change management policies may cover
  - The change request process and the information required to process each change request
  - The process used to analyse the impact and costs of change and the associated traceability information
  - The membership of the body that formally considers change requests
  - Software support (if any) for the change control process
- A change request may have a status as well as requirements
  - E.g., proposed, rejected, accepted, included...

26

Change Request Form

- Proposed changes are usually recorded on a change request form which is then passed to all of the people involved in the analysis of the change
- Change request forms may include
  - Date, Customer, Requester, Product including version
  - Description of change request including rationale
  - Fields to document the change analysis
  - Signature fields
  - Status
  - Comments

27

What Kind of Tool Do We Need?

Different companies will use different tools, which may or may not be tailored to the requirements management task

- Word processor (Microsoft Word with templates…)
- Spreadsheet (Microsoft Excel…)
- Industrial-strength, commercial RM tools
  - IBM/Telelogic DOORS, IBM Requisite Pro, Borland CaliberRM…
- Internal tools
  - GenSpec (Hydro-Quebec)…
- Open source RM tools
  - OSRMT: http://sourceforge.net/projects/osrmt
- Bug tracking tools (free or not)
  - Bugzilla…
- Collaboration tools (free or not)

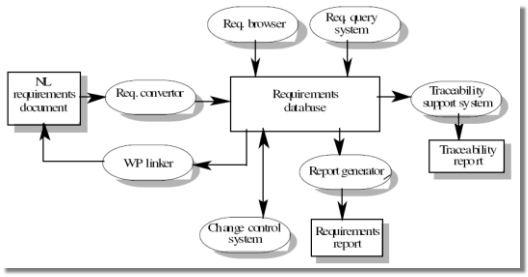
28

What Should We Look For in a Tool?

- Types/attributes for requirements and links
- Specifications and models
- Version and change management
- Database repository
- Traceability
- Analysis (impact, completeness, style, differences…)
- Automatic inspection of requirements (according to rules)
- Visualization and reports
- Requirements document generation
- Monitoring of requirements statuses
- Access control
- Import/export
- Communication with stakeholders
- Scripting language (for automation)
- Reuse of requirements, models, projects
- …

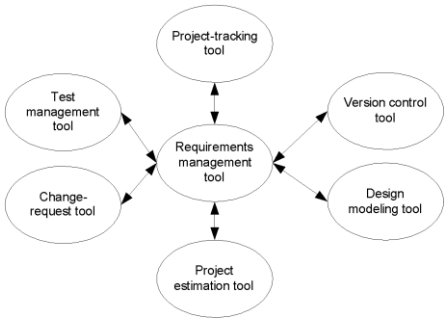
29

RM Tool Architecture – Example



30

Requirements Management Implies Integration!



31

TWiki Overview

- A generic Wiki tool (TWiki.org)
  - Promotes collaboration
  - Database-driven
  - Access and version control
  - Forms and queries
  - State-based workflows (processes)
  - Text and graphics
  - Lightweight, extensible (plug-in architecture)
- Example of Forms and Queries
  - Requirements: http://cserg0.site.uottawa.ca/twiki/bin/view/ProjetSEG/UCMNavRequirements
  - Library: http://cserg0.site.uottawa.ca/twiki/bin/view/UCM/UCMVirtualLibrary
  - Use Cases: http://cserg0.site.uottawa.ca/seg/bin/view/CSI4900/UseCases

32

TWiki for Requirements Management

Current Requirements

- Add a new requirement: Req Create
  (Requirement name format is ReqNameOfRequirement)

Name	Description	Type	Importance	Priority	Status	Dependencies	Date
Req Action Redo Many	jUCMNav SHOULD allow the user to redo many of the latest transformations or actions undone on the UCM model.	Functional	Optional	Low	Implemented	Req Action Undo Many, Req Action Redo One	13 Mar 2005 - 19:58
Req Action Redo One	jUCMNav SHOULD allow the user to redo the latest undone transformation or action on the UCM model.	Functional	Optional	Low	Implemented	Req Action Undo One	13 Mar 2005 - 20:01
Req Action Undo Many	jUCMNav SHOULD allow the user to undo many of the latest transformations or actions on the UCM model.	Functional	Optional	Medium	Implemented	Req Action Undo One	13 Mar 2005 - 20:01
Req Action Undo One	jUCMNav SHALL allow the user to undo the latest transformation or action on the UCM	Functional	Mandatory	Urgent	Implemented		13 Mar 2005 -

33

Twiki – Requirement Example

ProjetSEG > UCM Nav Requirements > **ReqActionUndoOne**

Edit this page | Attach a file | Printable version | More...

This is an important feature that may impact how the model is maintained. Does EMF help here?

-- DanielAmyot - 30 Jan 2005

When using GEF/EMF, every action performed creates a Command object. You create the Command and its inverse if you ever need to bring the model back to its original state. This command object is stored in some stack somewhere and the redo/undo functions are handled by the framework. Etienne's network editor already has this functionality. (I don't know how many commands are contained in the stack.

-- JasonKealey - 03 Feb 2005

Attach for Req	
Req Name:	Req Action Undo One
Description:	jUCMNav SHALL allow the user to undo the latest transformation or action on the UCM model.
Type:	Functional
Importance:	Mandatory
Priority:	Urgent
Status:	Implemented
Author:	DanielAmyot
Dependencies:	
Verification Approach:	JUnit
Test Cases:	Test Undo Redo

34

TWiki – Requirement Form Example

ProjetSEG > Quick Twiki Forms Facts > **FormForReq**

Edit this page | Attach a file | Printable version | More...

Name	Type	Size	Values	Tooltip message
Req Name	text	35		A Wiki name, starting with Req
Description	textarea	60x4		Use SHALL or SHOULD statements
Type	select	1	Functional, Non Functional, GUI, Platform, Goal	*Goal* is not a requirement but an objective
Importance	select	1	Mandatory, Optional, Future	*Future* means to consider beyond this project
Priority	select	1	Urgent, Medium, Low	
Status	select	1	Proposed, Approved, Rejected, Started, Implemented, Completed	*Completed* means implemented ***and*** tested
Author	text	35		Wiki name of the author (do not forget the Main. prefix)
Dependencies	text	60		List of requirement Wiki names (and optionally type of dependency)
Verification Approach	text	60		A few words about the strategy (or Wiki name)
Test Cases	text	60		List of test case Wiki names, if any

35

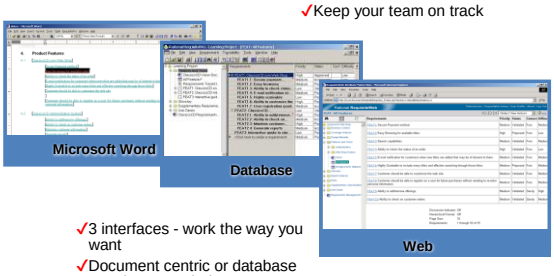
Using TWiki…

- We have:
  - Requirement types description with configurable statuses & attributes
  - Bidirectional links (WikiWords)
  - Configurable requests, filtering, reports
  - Access control and version management (showing differences)
  - Change management (again with forms, process, etc.)
  - Discussions, attachment of documents/images
  - Export (HTML)
  - Scripting language (Perl)
- But do we really have:
  - Graphical view of traceability?
  - Editable tables (à la Excel/Word)?
  - Baselines? Tool integration? Imports? Analysis?

36

IBM Requisite Pro

✓Keep your team on track



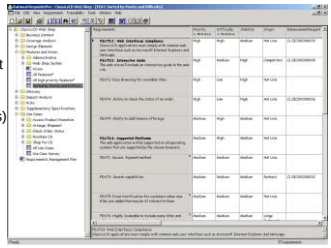
✓3 interfaces - work the way you want

✓Document centric or database centric - your choice

37

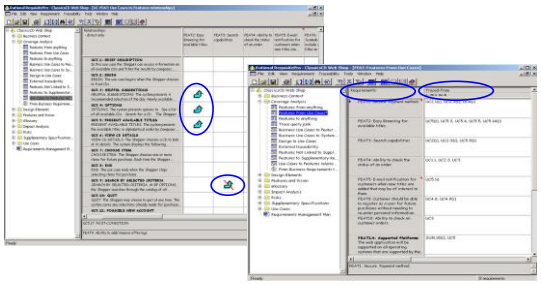
IBM Requisite Pro – Types, Attributes, and Views

- ✓ User defined requirement types
- ✓ User defined attributes
- ✓ User defined filters (views)
- ✓ Saved views



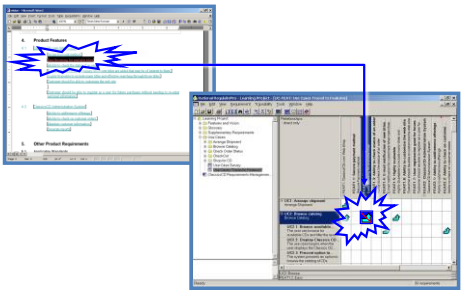
38

IBM Requisite Pro – Traceability



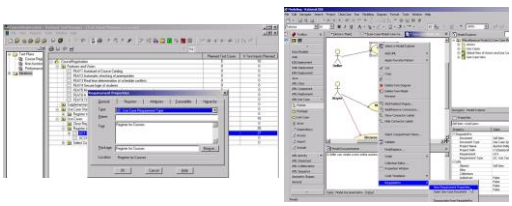
39

IBM Requisite Pro – Change Management



40

IBM Requisite Pro – Integration



- ✓ IBM Rational TestManager
- ✓ Testers view current state of requirements from their tool

- ✓ IBM Rational XDE and IBM Rational Rose, Rational Software Architect and Rational Software Modeler
- ✓ Developers view current state of requirements from their tool

41

END OF TOPIC # 11

-COMING UP!!!!!!
-Stakeholder Profiling
-Semester Exam
-Project Submission & Viva

42