How to Select a Requirements Management Tool: Initial Steps
Orlena (Olly) Gotel and Patrick Mäder

17th IEEE Requirements Engineering Conference (RE’09)
September 2, 2009
Which tasks can be supported by requirements management tools?

When should I use a tool for requirements management?

How can I identify the right tool for my project and my organization?

How can I optimize the tool support?

Which requirements management tool do you recommend?

No-one is using the requirements management tool – what do I do?
Motivation, objectives and assumptions

What exactly is Requirements Management (RM)?

Where do tools fit into a RM system?

Categories and capabilities of tools

Designing your RM system and selecting tools

Indicators of likely success or failure

Accolades and grumblefest
Motivation, objectives and assumptions

What exactly is Requirements Management (RM)?

Where do tools fit into a RM system?

Categories and capabilities of tools

Designing your RM system and selecting tools

Indicators of likely success or failure

Accolades and grumblefest
Which do you recommend?
Which do you recommend?
Laundry tablets

Remove stains from your washing

Four Which? Best Buys to pop into your machine. Which? review of pre-measured powder tablets and liquid sachets. Ariel, Bold, Daz, Fairy and Surf face the supermarket brands. How to remove tough stains from your laundry. What detergents are recommended for sensitive skin.

In our full report you can see...

**Best Buys**
Pick from our Best Buys to obtain better all-round cleaning.

**Don’t Buys**

**Full test results**
Discover how well the range of tablet and sachet detergents work.

Sign up to read the full Best Buy review
1 month trial for just £1.00

[http://www.which.co.uk]
Commercial Tools

Accept 360°  Accompia  Arcway  Cockpit  Agility  Agilo for Scrum  Aligned Elements  ARM  Blueprint  Caliber  Care  CaseComplete  CaseSpec  Comply Pro  Contour  Cradle  CRW  DocuBurst  DOORS  FeaturePlan  Focal Point  GatherSpace  Gmarc  iRise  IROA  Leap SE  Lighthouse  MKS  Integrity  MockupScreens  Objectiver  OptimalTrace  Pace  Pixref  Polaris  Rally  RaQuest  Raven  ReMa  Requirements Management Database  RequisitePro  RESDES  Rhapsody  ROA  Statestep  Teamcenter  TopTeam Analyst

Requirements Management Software

Accept 360  AnalysisPro  CORE  DOORS & DOORSrequireIT  GatherSpace  iRqA  Leap SE  Mac A&D and Win A&D objectIF  Optimal Trace  PixRef Pro  Qualidea  QFD  RaQuest  ReqItify  Requirement Tracing System  RMTrak  SoftREQ  Tiger PRO (free)  Tracer (free)  XTiE-Requirements Tracer  Active!Focus  Caliber-RM  Cradle  Enterprise Architect  GMARC  Jama Contour  Lighthouse RM  MKS Requirements  Open Source RM  PACE  Projectivity  Rally  Reconcile  Requirements Mgmt Database  Requisite Pro  RTM Workshop  Teamcenter  TopTeam Analyst  TRUEreq (free)

Which do you recommend?

[http://easyweb.easynet.co.uk/~iany]  

The Gartner Magic Quadrant

Source: Gartner (October 2005)

[http://www.gartner.com]
The Forrester Wave™

[http://www.forrester.com]
# INCOSE Requirements Management Tools Survey

<table>
<thead>
<tr>
<th>Tools:</th>
<th>CASE Snoop 8.0</th>
<th>CARE 3.2</th>
<th>Compuware Optimal Trace</th>
<th>CORE 5.1</th>
<th>Cradle 5.2</th>
<th>Envision VIP</th>
<th>GatherSpace</th>
<th>IBM Rational RequisitePro (updated 10 Oct 06)</th>
<th>KollabNet Editor 2005</th>
<th>Pace</th>
<th>ReQuest 3.1</th>
<th>RDT 3.0.2</th>
<th>RMTrak</th>
<th>RTM</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Capturing requirement identification</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
</tr>
<tr>
<td>1.1.1.1 Input document enrichment/analysis</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
</tr>
<tr>
<td>1.2 Automatic parsing of requirements</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>None</td>
<td>Full</td>
<td>Full</td>
</tr>
<tr>
<td>1.3 Interactive/semi-automatic requirement identification</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
</tr>
<tr>
<td>1.4 Manual requirement identification</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
</tr>
<tr>
<td>1.5 Batch-mode operation</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
</tr>
<tr>
<td>1.6 Requirement classification</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
</tr>
<tr>
<td>2. Capture System Element structure (if so, how? As document paragraphs? product structures? ...)</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
</tr>
<tr>
<td>2.1 Graphically capture systems structure</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
</tr>
<tr>
<td>2.2 Textual capture of system structure</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>None</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
</tr>
<tr>
<td>3. Requirements Flowdown</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>None</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
</tr>
</tbody>
</table>
MONDAY, JULY 30, 2007

Seilevel’s Requirements Management Tool Selection

Over the spring, we took on a new project to identify a best of breed requirements management tool to use when our customers are not using sufficient tools. Since the thing we do is requirements... well, you won't find it too surprising to hear that we defined requirements for a requirements management tool!

We successfully applied our vendor selection process to this project. The product management team identified actors, use cases, data types, functional and non-functional requirements. We ultimately used about 150 requirements in our selection process. To identify a vendor list, we used our knowledge of tools, our customer’s tool experiences and the INCOSE Requirements Management Tools Survey. A co-worker had previously evaluated the INCOSE survey list and eliminated many of the vendors. We did a slightly more in depth review including RequisitePro, Contour, Doors, Caliber, CaseComplete, FeaturePlan, CaseComplete and FeaturePlan. We narrowed it further to Doors, Caliber and RequisitePro based on feature richness.

I'll jump to the punch line and tell you we selected RequisitePro. They were our tool of choice and are currently piloting it on our newest projects.
What this mini-tutorial WON’T do
- Recommend a RM tool
- Focus on requirements traceability

What this mini-tutorial WILL do
- Try to summarize what is out there
- Suggest initial steps to help you figure it out for yourself
- Provide some advice for success in implementation

Primary audience
- Practitioners looking for a place to start
Why us?

- Olly:
  - Extensive RM tools survey in the early 1990s
  - Suffered RM tools first-hand in the later 1990s
  - Ongoing R&D on RE topics and tools

- Patrick:
  - Developed plug-ins for RM for tools
  - Recent RM tools survey
  - Ongoing practitioner interviews / worked for a vendor
- Motivation, objectives and assumptions
- What exactly is Requirements Management (RM)?
- Where do tools fit into a RM system?
- Categories and capabilities of tools
- Designing your RM system and selecting tools
- Indicators of likely success or failure
- Accolades and grumblefest
Language – where most problems originate!

- People use the following terms inconsistently:
  - Requirements management
  - Requirements engineering
  - Requirements development
  - Requirements traceability

If you are selecting a requirements management tool, agree what the term means!
For our purposes:

Based on [Wiegers 1999]
“The purpose of Requirements Management (REQM) is to manage the requirements of the project's products and product components and to identify inconsistencies between those requirements and the project's plans and work products.”

[CMMI 2006]
The requirements part?

1. “A condition or capability needed by a user to solve a problem or achieve an objective.

2. A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed document.

3. A documented representation of a condition or capability as in 1 or 2.”

[IEEE Glossary]
The management part?

- Process or practice:
  - “Organization and coordination of the activities of an enterprise in accordance with certain policies and in achievement of clearly defined objectives.”

- Person or persons:
  - “Directors and managers who have the power and responsibility to make decisions to manage an enterprise.”

[to lead by the hand]

WHY bother with RM?

- To help you maintain agreement on requirements throughout the development process

- To help you deliver systems that meet these requirements – confidence in customer satisfaction

WHO is it for?

- Project manager, product manager, quality manager, customer, marketing, engineering (business analyst, requirements engineer, architect, designer, developer, tester), financial department, legal, regulators – these all afford perspectives

And more…
WHAT is typically expected from RM?

- Better quality requirements
- Better ability to plan, estimate, allocate, track and control work
- Better ability to manage changing requirements
- Better ability to branch and backtrack
- Better project memory and continuity
- Better ability to reuse work
- Better ability to (demonstrably) meet contracts
- Better use of time etc.
WHAT RM can and can’t deliver

- Unambiguous, complete, correct requirements – NO! That’s the realm of writing better requirements, and performing effective reviews and validation

- Reduction in requirements-related defects – NO! That’s reliant on the quality of requirements development practices, so can still deliver the wrong requirements (GIGO)

- Useful analyses – YES! Completeness, coverage, compliance, risk, status, derivation, volatility, likely quality, gaps, criticality, change impact, V&V, complexity, failure probability, etc.

Is RM really just glorified housekeeping / gardening?

If you are selecting a requirements management tool, agree the scope of RM!
Great RM quotes from your RE colleagues …

Investing in putting bad requirements into management tools is a lot like rearranging the deck chairs on the Titanic

-- Ivy Hooks

If you are selecting a requirements management tool, look to the wider RE problem!
WHAT activities does RM involve?

- Obtaining and storing data in a shared place
- Augmenting these data
- Organizing and relating these data
- Accessing and reporting on these data
- Updating and changing these data, and their organization
- … all while preserving the integrity of these data and their inter and intra relationships
WHAT does RM involve?

- Obtaining and storing data in a shared place
- Augmenting these data with meta-data
- Organizing and relating these data: Structuring and establishing traceability
- Accessing and reporting on these data: Views, analytics, reports, export, controls
- Updating and changing these data, and their organization: Change management, propagation
- … all while preserving the integrity of these data and their inter and intra relationships: Versioning, baselining, impact analyses, notifications, maintenance, pruning to avoid entropy, refactor

Not just requirements, but other development assets, and in various forms – import, create, editors…
Motivation, objectives and assumptions

What exactly is Requirements Management (RM)?

Where do tools fit into a RM system?

Categories and capabilities of tools

Designing your RM system and selecting tools

Indicators of likely success or failure

Accolades and grumblefest
Fact

- All the core activities of RM can be done manually, but this can be tedious and error-prone

Reality

- RM tools can really help as you start to scale and distribute, and even automate many aspects, but they differ in the nature and extent of this support

Perception

- RM tool == requirements traceability tool

Fact

- Most focus on the support provided for traceability when selecting a tool and the resulting analyses this brings
Components of a RM system

People and Other Resources

Data Repository

Process

Techniques, Methods and Tools
People and Other Resources

Clear roles and responsibilities for undertaking the activities

Generally an underlying database: open, multiple media, multi-user, etc.

Process

Policies and procedures to weave people and activities together

Data Repository

How the various RM activities are to be performed and supported

Techniques, Methods and Tools
WHEN to use a tool to support your RM system

- Need to distribute, share and align: More than one person / site / organization doing the RE / development work
- Need to scale: Project has many requirements and / or multiple releases (i.e., need for release management)
- Need to diversify and reuse: Requirements are being used in many ways (i.e., product families)
- Need to know: Desire to improve quality, decisions and gather metrics
- Need to associate: More than one engineering step is necessary to transform the requirements into the desired result
- Need to alleviate: Staff are under-utilized with repetitive and administrative tasks
- Need to demonstrate: Contractual or legal reasons
- Need to maintain: Long project life expected, many customers
- Motivation, objectives and assumptions
- What exactly is Requirements Management (RM)?
- Where do tools fit into a RM system?
- Categories and capabilities of tools
- Designing your RM system and selecting tools
- Indicators of likely success or failure
- Accolades and grumblefest
Primary support

- Obtain/store – shared database, different artifact types, import from other tools, parsing and linguistic analyses, templates
- Augment – define attributes for manual / automatic capture
- Organize/relate – {structuring, grouping, hierarchy, flow-down, derived} requirements, manual and automated linking (within and external to tool)
- Access/report – security and set-up, analyses, search, sort, filter, navigate, visualize in different ways, export
- Update/change – editing, define / support change control process, end-to-end traceability, review process
- Preserve integrity – define baselines, versioning, audits and history, assess impact, automatic updates, suspect links, notifications, rollback, support approvals process
Basic architecture of a RM tool

- Import
- Database
- Editor
- Export
- Report Generator
Enabling support

- Process definition
- Workflow
- Teamwork
- Collaboration, Coordination, Communication
- Admin: accounts, backup, etc.

... these all enable the fundamental activities of RM, help to differentiate and add potential value!
Categories of tool

- General-purpose tools configured to support RM tasks
- Special-purpose tools dedicated to RM tasks
- Life-cycle tools with RM capability

General purpose RM tools and domain-specific RM tools
General-purpose tools

- Hand-configured to allow previously manual and paper-based RM activities to be carried out

**Pros:**
- Widely and readily available
- People know how to use them
- Flexible
- Okay for small projects
- May help you progress with your RM learning adventure!

**Cons:**
- Appear cheap, but could incur high start-up cost
- Difficult to change / maintain
- Unpredictable support – do not provide the necessary basis functionality for RM
- Avoid on sizable, distributed or long-term projects
Examples: General-purpose tools

- Text editors
- Graphic editors
- Flow chart tools
- Spread-sheet tools
- Databases
- Wiki (with RSS)
- ...
Special-purpose tools

- Typically centered around DB management system, with dedicated tools for documenting, parsing, editing, grouping, decomposing, linking, organizing, partitioning and managing requirements

- Similar structure (GUI, database, editor, import / export)

**Pros:**
- Focuses on RM
- Added value RM
- Varies in degree of support for RM – emphasizing different activities
- RT often a by-product of work process
- Specific analyses, reports…
- Often customizable
- Dedicated vendor support on RM?

**Cons:**
- Can be rigid, with fixed RM process, or may not help you define your own
- RT can become focal to detriment of wider activities
- Standalone, so integration with other tools can vary, meaning full end-to-end traceability can be difficult
- Seductive – you assume it does your work!
Examples: Special-purpose tools

- Borland Caliber RM
- IBM Rational Requisite Pro
- IBM Telelogic DOORS
- ...

Life-cycle tools

- Provide support for many (or all phases) of systems lifecycle (analysis, design, coding, testing and management)

- Not usually specialized for RM, but provide capability

- Very different RM functionality between these tools, some do not cover the main RM tasks, others provide almost the functionality of a special-purpose tool

Pros:
- Leverage full life-cycle data, so end-to-end traceability possible (in theory) with full application life-cycle coverage
- Avoids gaps between artifacts (e.g., requirements–test or requirements–modeling)
- Can be open environments

Cons:
- Different focus, RM just one part, more a side-effect
- Not always so sophisticated support
- May need configuring / integration
- Buy in to full SDLC approach
Examples: Life-cycle tools

- Modeling tools (UML: use case, behavioral diagrams, structural diagrams; SysML: requirements diagram)
  - Examples: ARTiSAN Studio + ReqTify, Sparx Enterprise Architect

- Test management tools

- Bug- / issue-tracking
  - Example: JIRA, Polarion
Maybe you need multiple tools?

- Advantages of using fewer tools:
  - Little or no integration of artifacts
  - Consistency problems for different artifacts
  - Fewer tools to learn and handle – developer is familiar with tool and might not want another one

- Advantages of using multiple tools:
  - Very specific solutions for different phases of development
  - If a dedicated RE / BA role – better to have own special-purpose tool?
  - If a contractor, you may be forced to
How a RM system may fit into a wider SDLC system

- Integrate with design (modeling tool) or code (IDE)
- Integrate with test management tools
- Link to directives and external documents
- Integrate with project management tools
- Integrate with the customer’s / sub-contractor’s RM system
  - Import, export, access rights, traceability
  - Share data among different RM tools
<table>
<thead>
<tr>
<th>Feature</th>
<th>Special-purpose tool</th>
<th>Life-cycle tool</th>
<th>General-purpose tool</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model-based management of requirements with attributes</td>
<td>DOORS</td>
<td>UML tool</td>
<td>Wiki</td>
</tr>
<tr>
<td>Organizing requirements</td>
<td>(+) reduced number</td>
<td>(+)</td>
<td>(+)</td>
</tr>
<tr>
<td>Version management of single requirements</td>
<td>(+) usually package level</td>
<td>+</td>
<td>–</td>
</tr>
<tr>
<td>Configuration management / base-lining</td>
<td>+</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Multi-user support</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Traceability management</td>
<td>(+) often with limitations</td>
<td>(+) with limitations, RSS change propagation</td>
<td>–</td>
</tr>
<tr>
<td>Change control</td>
<td>(+) usually document or package level</td>
<td>(+)</td>
<td>–</td>
</tr>
</tbody>
</table>
- Motivation, objectives and assumptions
- What exactly is Requirements Management (RM)?
- Where do tools fit into a RM system?
- Categories and capabilities of tools
- Designing your RM system and selecting tools
- Indicators of likely success or failure
- Accolades and grumblefest
ARTiSAN Studio, Borland CaliberRM, Borland Together, IBM Rational Rhapsody, IBM Rational RequisitePro, IBM Rational Rose Professional C++,

Tools can EITHER make your RM system more efficient and effective OR lead you to no end of problems… it is up to you!
So many tools -- which tool to use and when?
Let’s not start here, but step back…
10 (not so simple – but important) steps

1. Agree on what RM is
2. Agree there is a problem to be tackled
3. Understand the problem to be tackled
4. Secure top-level commitment to tackle it
5. Identify stakeholders and secure their buy-in
6. Ensure RM requirements are stakeholder and context-driven
7. Agree scope of the RM solution and design RM system
8. Assess the value of tools in this RM system
9. Make or buy – either way, do your homework
10. Treat it as a serious process improvement project
Generic problem solving, RE and tool selection process (1)

Understand the opportunity

- Business case development & approval
- Planning & tool selection
- Implementation

Agree there is a problem to be tackled

Identify stakeholders

RM process exists?

- Not mutually exclusive
- Yes
  - Interview stakeholders
  - Interview process owners

"As Is"

Document current RM process
- Identify pain points / issues
- Identify goals

"To Be"

Design concept of new RM process

Document opportunity using new process assuming tools to support

Agree the business driver – the problem you are trying to fix or the value you are trying to gain
Generic problem solving, RE and tool selection process (2)
Generic problem solving, RE and tool selection process (3)

1. Understand the opportunity
2. Business case development & approval
3. Planning & tool selection
4. Implementation

Elaborate Requirements for RM:
- Needs analysis
- Context
- NFRs
  - Security, Scalability, Reliability, Usability, Accessibility
  - Portability, Performance, etc.
- Constraints
- Change management

Design RM system
Tool selection
Tool procurement
Generic problem solving, RE and tool selection process (3a)

- Understand the opportunity
- Business case development & approval
- Planning & tool selection
- Implementation

Elaborate Requirements for RM:
- Needs Analysis
- Context

- Explore stakeholder constituencies: roles, tasks, goals & information needs – create scenarios of use
- Understand context: organization type, SDLC process, systems & resources, SPI culture, project characteristics – establish system scope & boundaries

Design RM system:
- What changes & what stays?
  - RM activities to emphasize
  - Roles & responsibilities to undertake them
  - Process to weave them together
- Traceability analyses & information model
  - Identify bridges & barriers
Identifying stakeholders – try using Ian Alexander’s approach (see [Alexander and Beus-Dukic 2009])
An initial attempt

RM problems are not uniform.
They differ according to stakeholder roles.
Direct and indirect stakeholders.
Myriad roles.

What are their goals and tasks?
What decisions can RM help them with?
Assess value.

Myriad roles.
Stakeholders and scenarios for RM

- Developer
  - View assigned open requirements
  - Trace requirements
  - Check quality of specification
  - View untested requirements
  - Get specification

- Product Manager
  - View requirements status
  - View requirements with structural impact
  - Provide implementation status

- Tester
  - Estimate impact of changing requirement
  - Find requirement to test

- Customer
  - Find responsible for requirement
  - Review requirements
  - Provide needs

- Quality Administrator
  - Find responsible for requirement

- Sub-contractor
Detailed use case

1. Lock or baseline all requirements under review
2. Identify and inform reviewers
3. Provide access to the requirements for each reviewers
4. Let reviewer comment on each requirement
5. Perform changes to requirement
6. Store review board decision on each requirement
7. Provide aggregated view with all comments on each requirement
8. Check each requirement has been commented or viewed by each reviewer
Generic problem solving, RE and tool selection process (3)
Generic problem solving, RE and tool selection process (3b)

Vendor details:
- Company background
- Financials
- Capabilities & resources
- Strategy

Product details:
- Market share
- Roadmap & stage in lifecycle
- SDLC paradigm
- Features
- Platform
- Integration & extensibility

Criteria:
- Ranked requirements & scenarios
- Context fit, constraints & prior knowledge

Implementation:
- Project plan – prep, install, ongoing
- Team
- SLA – training, consulting...
- TCOO – initial, operating, recurring...

Reference customers:
- Challenges
- Lessons

Unlikely to be a perfect fit – can you configure the tool, can you customize it?
Why it can be difficult to evaluate tools

- Getting specific information regarding the tool features (e.g., directionality of traceability links)
- Getting a license
- Duration of license
- Getting the tool installed
- Understanding the underlying working and process assumption in the different tools in a short evaluation time
Tool quality indicators

- Does the website communicate?

- Does the documentation reflect the latest tool version?

- How often do major / minor updates appear?
  - Is a concrete change log for each version available, showing what was added, fixed, omitted?

- How reactive and useful is the support? (Are there traceable tickets for discovered problems?)

- Installation process: Can you get the tool working without additional support?
Tool quality indicators (developing own plug-ins)

- Is documentation for the tool API available?
  - Does the API documentation reflect the latest state?
- Are examples available?
- Does an active user community exist that can support you during plug-in development?
Generic problem solving, RE and tool selection process (4)
Guidance on roll out

- Installation
- Configuration
- Customization
- Data-migration
- Training
- Pilot project
- Ongoing support
Installation

- Server application
- Database
- Clients application
- Existing plug-ins and integration with other tools
Configuration: Structure

- Hierarchies:
  - Documents, modules, packages (Which parts shall be handled individually? What requirements types exist?)
  - Initial structure within documents
- ID system and structure per document (if possible)
- Attributes for requirements (enumerations, default values)
Configuration: Usage and integration

- Roles, views and access rights
- Versions and baselining (when, who)
- Reports and exports (e.g., reviews, requirements for subcontractors)
- Integration with other tools (set-up relation to subsequent models, e.g., UML models, tests, etc.)
Configuration: Traceability Information Model

- Specification of the required traceability for a project according the intended usage – what questions do you need to answer
- Each relation defines a permitted trace in the project
- Minimum count of required relations as multiplicities
- Types of relations and roles of artifacts
- Dependency between artifacts

![Diagram of Use Case and Test Case relationships]
Example of a Traceability Information Model
Customization

- Should be used cautiously (because of effort and costs) – find out whether the needed functionality is really not available!
- Might need evolution after update of the main tool due to API, database or functional changes
- Often difficult to create a general and final solution, so constantly triggering enhancements and improvements to the functionality
Data-migration

- Existing importer, customized importer, manually

Problems:

- Migrating the history of artifacts / requirements is difficult
- Different concepts for traceability
- Constraints to allowed ID’s, attributes…
Training

- Training on the tool, but also on the (new) RM process with the tool

Pilot project

- Goal: evaluation and training in the field with real data
- Constant support and consulting for the users
- Feedback rounds and reviews (control loop to improve process, tool support, tool configuration) $\rightarrow$ likely to trigger configuration changes and additional customization
Ongoing support: “The story is not finished …”

- Provide detailed and up-to-date information about the company’s RM process (with roles and responsibilities) and the tool set-up for new team members
- Provide project templates and support for new projects
- Get feedback from users on a regular basis
- Do regular reviews of tool-support and process
- Analyze completed projects to get information about the success of process and tool
- Decide whether to propagate improvements immediately to all projects or just for new projects
10 (not so simple – but important) steps

1. Agree on what RM is
2. Agree there is a problem to be tackled
3. Understand the problem to be tackled
4. Secure top-level commitment to tackle it
5. Identify stakeholders and secure their buy-in
6. Ensure RM requirements are stakeholder and context-driven
7. Agree scope of the RM solution and design RM system
8. Assess the value of tools in this RM system
9. Make or buy – either way, do your homework
10. Treat it as a serious process improvement project
Motivation, objectives and assumptions

What exactly is Requirements Management (RM)?

Where do tools fit into a RM system?

Categories and capabilities of tools

Designing your RM system and selecting tools

Indicators of likely success or failure

Accolades and grumblefest
Key success factors

- Management sponsorship, leadership and buy-in of team
- Fits your process (designed, communicated, shared)
- Systemic solution to RM (linking people, process and tools) as part of your wider SDLC
- Goals clear, with metrics to mitigate problems early
- Prepared environment: people trained, resourced, roles and responsibilities agreed, mentors, joint ownership, shared commitment (process improvement a goal)
- Incentives to do – must ease job or drive compensation
- Easy introduction and progressive change, prepared to navigate initial performance dip – treated as a project
- Credibility (current and accurate) -- use is a virtuous or vicious cycle – so clean up!
WHO is responsible for RM?
Expect problems when…

- You think RM is going to solve ALL your requirements-related problems -- you need to couple it with RD and SDLC process
- You did not articulate your goals and design a RM process to satisfy them – you expected the tool to provide this and do the hard work for you
- You selected a tool based on it having the most features rather than its support for your valued scenarios and context
- You started managing your requirements too late – as needed
- Representatives don’t use it as were not involved in decisions
- You did not think WHOLE team and how to handle the integration of all SDLC assets through life
- The workload ramped up unexpectedly, so you cut corners
- You miscalculated the total cost of ownership, funds dry up
- Results of RM analyses are used to witch hunt
More great RM quotes from your RE colleagues …

Requirements management is like flossing – everyone knows they should do it, but very few actually do because it’s far from something to get excited about -- Jama Software

If you are selecting a requirements management tool, think motivation and incentives!
- Motivation, objectives and assumptions
- What exactly is Requirements Management (RM)?
- Where do tools fit into a RM system?
- Categories and capabilities of tools
- Designing your RM system and selecting tools
- Indicators of likely success or failure
- Accolades and grumblefest
What kind of tools exist to support the requirements management process?

Which tasks can be supported by requirements management tools?

When should I use a tool for requirements management?

How well do the different categories of tool support the tasks of requirements management?

How can I optimize the tool support?

How can I identify the right tool for my project and my organization?

Which requirements management tool do you recommend?

No-one is using the requirements management tool – what do I do?
How did you select a RM tool?

Are you using the right RM tool?

Success stories?

Horror stories?

Tactics to turn things around? Advice?

The ‘ideal’ RM tool?
Olly Gotel
Pace University
New York City, USA
olly@gotel.net
www.gotel.net

Patrick Mäder
Ilmenau Technical University
Germany
patrick.maeder@tu-ilmenau.de
www.patrickmaeder.de