GETTING THE MOST OUT OF REQUIREMENTS MANAGEMENT TOOL

REQUIREMENTS EXPERT KARL WIEGERS EXPLAINS THE BENEFITS OF AN RM TOOL

> Most project teams create software requirements specification written in natural language text to contain their functional and nonfunctional requirements, business requirements, use-case descriptions, and so forth. A document-based approach to storing requirements has numerous limitations, including the following:

- It's difficult to keep the documents current and synchronized.
- Communicating changes to all affected team members is a manual process.
- It's not easy to store supplementary information (attributes) about each requirement.
- It's hard to define links between functional requirements and other system elements.
- Tracking requirements status is cumbersome.
- Concurrently managing sets of requirements that are planned for different releases or for related products is difficult.
- Reusing a requirement means that the business analyst must copy the text from the original software requirements specification (SRS) into the SRS for each other system or product where the requirement is to be used.
- It's difficult for multiple project participants to modify the requirements safely, particularly if the participants are geographically separated.
- There's no convenient place to store proposed requirements that were rejected and requirements that were deleted from a baseline.

[1] Adapted from Karl E. Wiegers, **Testing the Requirements,** Microsoft Press, 2006.

A commercial requirements management tool that stores information in a multiuser database provides a robust solution to these restrictions. Such products let users import requirements from source documents, define attribute values, filter and display the database contents, export requirements in various formats, define traceability links, and connect requirements to items stored in other software development tools.

More than three dozen such tools are on the market today. They range from simple Web-based structures for storing requirements information to powerful multiuser, Web-enabled products with rich feature sets that can handle extremely large projects. I won't attempt to describe the capabilities of all these tools here or to provide specific recommendations. Descriptions and comparative information about many tools can be found online. Avoid the temptation to develop your own RM tool or to cobble together general-purpose office automation products in an attempt to mimic the commercial products. This initially looks like an easy solution, but it can quickly overwhelm a team that doesn't have the resources to build the tool it really wants.

Note that I classify these products as requirements management tools, not requirements development tools. They won't help you identify your prospective users or elicit the right requirements for your project. However, they provide a lot of flexibility in managing changes to those requirements and using the requirements as the foundation for design, testing, and project management. Nor do these tools replace a defined process that your team members follow to elicit and manage its requirements. Use a tool when you already have an approach that works but that requires greater efficiency; don't expect a tool to compensate for a lack of process, discipline, experience, or understanding.

Many of these tools aren't cheap. The high cost of requirements-related problems can justify your investment in them, though. Recognize that the cost of a tool is not simply what you pay for the initial license. The cost also includes maintenance fees and periodic upgrades, annual subscription costs if the product is delivered in the form of software as a service, and the costs of installing the software, performing administration, obtaining vendor support and consulting, and training your users. Your cost-benefit analysis should take into account these additional expenses before you make a purchase decision.

This paper presents several benefits of using requirements management tools and identifies some general capabilities you can expect to find in such a product. This paper offers some suggestions about how to sensibly integrate a tool into your business and get the maximum benefit from your investment in an RM tool, preventing it from becoming expensive shelfware.

BENEFITS OF USING A REQUIREMENTS MANAGEMENT TOOL

Even if you do a magnificent job specifiying your project's requirements, automated assistance can help you work with these requirements as development progresses. A requirements management tool becomes most beneficial as time passes and the team's memory of the requirements details fades. The following sections describe some of the tasks such a tool can help you perform.

1. Manage Versions & Changes.

Your project should define one or more requirements baselines, specific collections of requirements that are allocated to a particular release or iteration. Some requirements management tools provide flexible baselining functions. The tools also maintain a history of the changes made to every requirement. You can record the rationale behind each change decision and revert to a previous version of a requirement if necessary. Some of the tools contain a change-proposal system that links change requests directly to the affected requirements.

2. Store Requirements Attributes.

You should record several descriptive attributes for each requirement. Everyone working on the project must be able to view the attributes, and selected individuals will be permitted to update attribute values. RM tools generate several system defined attributes, such as the date a requirement was created and its current version number, and they let you define additional attributes of various data types. Thoughtful definition of attributes allows stakeholders to view subsets of the requirements based on specific combinations of attribute values. For instance, you might ask to see a list of all the requirements originating from a specific business rule so that you can judge the consequences of a change in that rule. One way to keep track of the requirements that are allocated to the baselines for various releases is by using a Release Number attribute.

3. Facilitate impact Analysis

The tools enable requirements tracing by letting you define links between different types of requirements, between requirements in different subsystems, and between individual requirements and related system components (for example, designs, code modules, tests, and user documentation). These links help you analyze the impact that a proposed change will have on a specific requirement by identifying other system elements the change might affect. It's also a good idea to trace each functional requirement back to its origin or parent so that you know where every requirement came from. Some of the tools let you establish traceability links between requirements in the database and objects stored in other, third-party tools, such as problem reports, change requests, design model objects, source code files, and project task lists.

4. Track Requirements Status

Collecting requirements in a database lets you know how many discrete requirements you've specified for the product. Tracking the status of each requirement during development supports the overall status tracking of the project. A project manager has good insight into project status if he knows that 55 percent of the requirements committed to the next release have been verified, 28 percent have been implemented but not verified, and 17 percent are not yet fully implemented.

5. Control Access

The requirements management tools let you define access permissions for individuals or groups of users and share information with a geographically dispersed team through a Web interface to the database. Communicate with stakeholders. Some tools permit team members to discuss requirements issues electronically through threaded conversations. Automatically triggered email messages notify affected individuals when a new discussion entry is made or when a specific requirement is modified.

6. Reuse Requirements

Storing requirements in a database facilitates reusing them in multiple projects or subprojects. Requirements that logically fit into multiple parts of the product description can be stored once and referenced whenever necessary to avoid duplicating requirements.

SOME REQUIREMENTS MANAGEMENT TOOL CAPABILITIES

Commercial requirements management tools let you define different requirement types (sometimes called classes), such as business requirements, use cases, functional requirements, hardware requirements, and constraints. This lets you differentiate individual objects that you want to treat as requirements from other useful information contained in the SRS. The tools provide strong capabilities for defining attributes for each requirement type, which is a great advantage over the typical document-based SRS approach.

Virtually all the tools have requirements traceability features that let you define links between objects of two requirement types, or even within the same requirement type. Many requirements management tools integrate with Microsoft Word to some degree. The higher-end tools support a rich variety of import and export file formats. Several of the tools let you mark text in a Word document to be treated as a discrete requirement. Some tools can parse documents in various fashions to extract individual requirements and load them into the database.

The tools often support hierarchical numeric requirement labels, in addition to maintaining a unique internal identifier for each requirement. These identifiers typically consist of a short text prefix that indicates the requirement type—such as UR for a user requirement—followed by a unique integer. Some tools provide efficient displays to let you manipulate the hierarchical requirements tree.

Output capabilities from the tools include the ability to generate a requirements document, either in a user-specified format or as a tabular report. Several tools let you define an SRS template in Word and then populate this template with information it selects from the database according to user-defined query criteria to produce a customized specification document. An SRS, therefore, is essentially a report generated from selected database contents.

Other features include the ability to set up user groups and define permissions for selected users or groups to create, read, update, and delete projects, requirements, attributes, and attribute values. Several of the products let you incorporate nontextual objects such as graphics and spreadsheets into the requirements repository. Some tools also include learning aids, such as tutorials or sample projects, to help users get up to speed.

Any of these products will move your requirements management practices to a higher plane of sophistication and capability. However, the diligence of the tools' users remains a critical success factor. Dedicated, disciplined, and knowledgeable people will make progress even with mediocre tools, whereas the best tools won't pay for themselves in the hands of unmotivated or ill-trained users. Don't write a check for a requirements management tool unless you're willing to respect the learning curve and make the time investment. Because you can't expect instantaneous results, don't base a project's success on a tool you're using for the first time. Gain some experience working with the tool on a pilot project before you employ it on a high-stakes project.

SELECTING A TOOL

Choose a tool based on the combination of platform, pricing, access modes, and organizing structure that best fits your development environment and culture. When selecting a tool, think about whether a database-centric or document-centric paradigm will be more effective for your organization. In a document-centric approach, you connect your word processing document to the tool, identify specific elements in the document as being discrete requirements, and import those into the tool either automatically or manually. Once those requirements are stored in the tool's database, you can manipulate them in the usual ways: assign attributes, define traceability links, and so forth. However, you need to keep the database contents synchronized with the document contents. This synchronization can get clumsy, but it's comfortable for people who are accustomed to working with documents. In addition, supplemental information, such as graphics and tables, should remain stored in the word-processing document if the tool can't handle those kinds of objects. This means that users of the requirements must go to multiple locations to get all the information they need about a specific portion of the new product.

The database-centric tools dispense with documents entirely and just treat the contents of the repository as the requirements collection. The more powerful products allow you to store various types of objects in the database, including graphics, as well as maintaining links to objects stored in documents or other files. An SRS then becomes a report generated from the database according to selected query and filtering criteria. The tools that are designed with a database at the center eliminate the clumsy synchronization between document and database that the document-centric tools demand.

When you go through your selection process, begin by defining your organization's requirements for an RM tool. Identify the capabilities that are most significant to you, such as the other tools with which you'd like the product to integrate. Decide whether you want to continue using documents to contain some of your requirements information or whether you prefer to store all the information in a database. The following selection procedure might be helpful:

1. List ten to fifteen factors that will influence your selection decision. Include subjective categories such as tailorability, as well as the efficiency and effectiveness of the user interface. Cost will be a selection factor, of course, but evaluate the tools initially without considering their cost.

2. Distribute one hundred points among the selection factors that you listed in step 2, giving more points to the more important factors.

3. Obtain current information about the available requirements management tools, and rate the candidates against each of your selection factors. Scores for the subjective factors will have to wait until you can actually work with each tool. A vendor demonstration can fill in some of the blanks, but the demo will likely be biased toward the tool's strengths. A demo isn't a substitute for wrestling the product yourself for several hours.

4. Calculate the score for each candidate based on the weight you gave each factor to see which products appear to best fit your needs.

5. Solicit experience reports from other users of each candidate product, perhaps by posting queries in online discussion forums, to supplement your own evaluation and the vendor's literature, demo, and sales pitch. Looking at the vendor's support forum is a way to see how frustrated current users appear and what sort of issues they struggle with.

6. Obtain evaluation copies from the vendors of your top-rated tools. Define an evaluation process before you install the candidates to make sure you get the information you need to make a good decision.

7. Evaluate the tools by using a real project, not just the tutorial project that comes with the product. After you complete your evaluations, adjust your rating scores if necessary and see which tool now ranks highest.

8. To make a decision, combine the ratings, licensing costs, and ongoing costs with information on vendor support, input from current users, and your team's subjective impressions of the products.

SUCCESS FACTORS FOR USING AN RM TOOL

Buying a tool is easy; changing your culture and practices to accept the tool and take best advantage of it is much harder. Most organizations already have a comfort level with storing their requirements in word processing documents. Changing to an online approach requires a different way of thinking and working. Keep the tips in this section in mind as you plan how to get the maximum return on your RM tool investment.

Write Good Requirements First

It's important to remember that these are requirements management tools, not requirements development tools. They won't help you define your business objectives, scope your project, identify the right user representatives, ask them the right questions, or write good requirements. The tools are not a substitute for effective requirements development processes and techniques. However, they will help you manage and track whatever information you store in them. It's a classic case of garbage in, garbage out.

Therefore, I do not recommend that organizations even pilot the use of a tool until the business analysts can already write good requirements. If your biggest problems are with eliciting and writing clear, high quality requirements, these tools won't help you. I've seen companies gain a false confidence in the quality of their requirements because the requirements were nicely stored in a database, well organized, and accessible through handsome reports. Nice looking but poor quality requirements don't help you much.

Expect a Culture Change

Organizations that are accustomed to storing requirements in documents already have mechanisms in place for creating, reviewing, approving, storing, distributing, and modifying those documents. A requirements management tool brings a significant new paradigm to these organizations.

Requirements are ethereal enough already. Being able to print the requirements gives them a slightly more tangible feel. But storing them in the deep, dark recesses of a database makes them even less tangible. It seems difficult for organizations to completely make the transition from the familiar document approach to storing requirements only in a database. One trap is investing effort into putting requirements in the tool while still retaining the original documents as the master requirements location. Your goal is to have all stakeholders regard the tool as the ultimate source of the current requirements for their project and not rely on the original requirements documents. This culture change will demand gentle pressure relentlessly applied by the managers and change agents in the organization to steer the organization to the new ways of thinking and working.

To accelerate the movement from a document-based paradigm to the use of the tool, set a date after which the tool's database will be regarded as the definitive repository of the project's requirements. After that date, requirements residing only in word-processing documents won't be recognized as valid requirements.

Don't Create Too Many Requirement Types or Attributes

Most requirements management tools allow you to define a variety of requirement type. Figure 1 suggests some requirements types to consider defining. You might identify an owner for each requirement type, who will have the primary responsibility for managing the database contents of that type. Each type can have its own set of attributes, pieces of data associated with that requirement type. For instance, you could define a use case requirement type that contains the components found in a typical use case template. You could also create a functional requirement type with a different set of attributes. Some attributes to consider for functional requirements are author, priority, status, origin, release number, validation method, rationale, and current owner. The tool will create and update certain attributes automatically, such as the creation date or date of last change. Other attributes are user-defined.

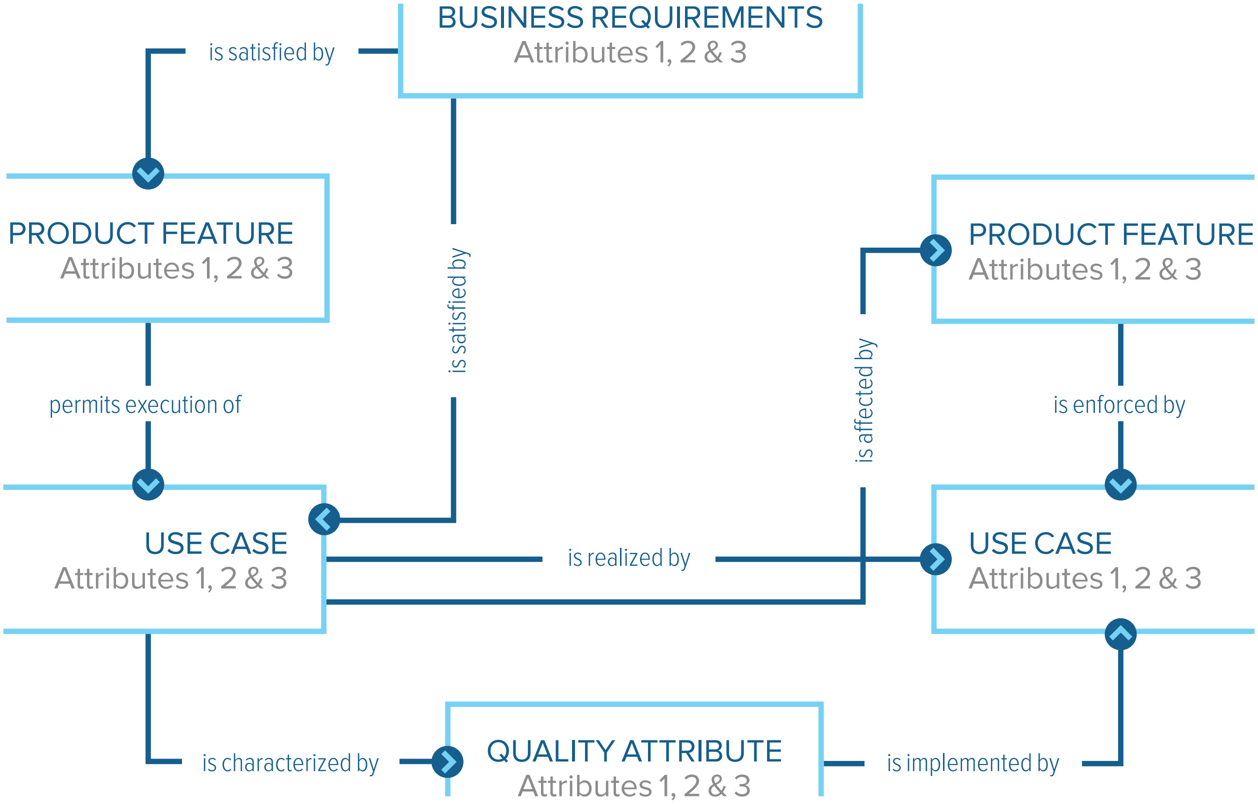

Figure 1. Some possible requirement types and potential traceability links

The arrows in Figure 1 illustrate traceability links that you could define to record the logical connections between different types of requirements stored in the tool. All of these requirements types, attributes, and traceability connections are potentially valuable. However, don't define any more of them than you really expect to populate and use because it takes effort to create and maintain them. Your team should select the information and traceability links that add value to its projects, and they should be diligent about storing that information and keeping it current.

It's easy to get carried away with designing the requirements database contents instead of thinking about how your team members will actually use the tool and the information stored in it. Instead of defining more attributes than you can manage, I'd rather see you define just three or four attributes initially, populate them, and take good advantage of the data. Priority, status, release number, and rationale comprise a good starter set of attributes for functional requirements.

Train the Tool Users

Although some requirements management tools are inexpensive, the high-end products can represent a significant financial investment. The team members must learn how to use your chosen tool appropriately and efficiently, so don't skimp on training. Your team members are smart, but it's better to train them than to expect them to figure out how best to use the tool on their own. They can undoubtedly deduce the basic operations, but they won't learn about the full set of tool capabilities and how to exploit them efficiently. Spending money on training and support after you've bought expensive licenses can be tough to sell to your managers. But if the participants don't know how to use the tool effectively, you won't get a suitable return on the investment.

For the smoothest transition, assign a tool advocate, a local enthusiast who learns the tool's ins and outs, mentors other users, and sees that it gets employed as intended. Begin with a pilot application of the tool on a noncritical project. This will help the organization learn how much effort it takes to administer and support the tool. The advocate will manage its use in the pilot and then train and mentor others to support the tool as other projects adopt it.

Assign Responsibilities

Someone must be responsible for the care and feeding of both the tool and the information stored in it. These are reasonable tasks for a business analyst, although the tool administration and content management functions could be split between different people. But the BA won't be the only person working with the information. You might need to give certain individuals authority to update attributes or add traceability data during the project. If these responsibilities are not made clear and accepted by the team members, important work won't get done. This degrades the quantity, quality, and value of the data stored in the tool.

For instance, it takes little effort—but considerable discipline—to accumulate traceability information as the software development work is being done. In contrast, it's very expensive or even impractical to assemble all the traceability data at the end of the project. All team members who are in a position to generate traceability data (such as developers and testers) must agree to record the traceability links as they do their work.

Time Activities Sensibly

You need to be thoughtful about when in your requirements development process you perform various activities with the RM tool. Don't try to capture requirements directly in the tool during the early elicitation workshops. As the requirements begin to stabilize, though, storing them in the tool makes them visible to the workshop participants for review and refinement. Also, don't define traceability links until the requirements stabilize, as when you define a baseline for a particular subset targeted at a particular iteration or release. Otherwise, you can count on doing a lot of work to revise the links as requirements continue to change.

Take Advantage of Tool Features

I know of one organization that exported all the requirements for a huge five-year project from their specification documents into a high-end RM tool. They defined countless traceability links among the various types of requirements stored in the tool. The only thing they did with all the data, though, was to generate hefty traceability reports. As it happened, no one in the organization actually used these reports. The analysts didn't exploit the other features this powerful tool provided, and the developers still relied on paper specifications as the definitive source of requirements. The investment this organization made in acquiring, installing, configuring, and populating its RM tool didn't yield a meaningful return.

Conversely, I know of projects that stored their requirements in a tool but didn't take advantage of any capabilities the tool offered for managing those requirements. One of the strongest arguments for requirements management tools is having the ability to define traceability links. The more robust requirements management products even allow analysts to establish such links to objects stored in other tools, such as to design elements stored in a modeling tool, code segments in a version control tool, and tests in a test management tool. If you don't use such features, it diminishes the value of keeping the requirements in a database.

The tools also let you define groups and individuals with different permission levels to identify who can read, create, and modify the contents of the database. Access controls are an important consideration for companies that have employees in multiple countries. These companies must be careful not to inappropriately expose sensitive technology and data to individuals who do not have the right to see that information. Take advantage of these tool capabilities to ensure that all—and only—the right people—can access the requirements.

Invest the Effort

Recognize that it will take effort to load a project's requirements into the database, define attributes and traceability links, keep the database's contents current, define access groups and their privileges, and train users. Management must allocate the resources needed for these operations. Commit to actually using your RM tool instead of letting it sit idle.

As the use of a requirements management tool becomes ingrained in your culture, project stakeholders will begin to regard requirements as life cycle assets, just as they do code. The team will discover ways to use the tool to speed up the process of documenting requirements, communicating them, and managing changes to them. Remember: even the best tool can't compensate for an ineffective requirements development process. The tool won't help you scope your project, identify users, talk to the right users, elicit the right requirements, or write good requirements. And it doesn't matter how well you manage poor requirements.

ABOUT KARL WIEGERS

Karl has provided training and consulting services worldwide on many aspects of software development, management and process improvement. He has authored 5 technical books, including Software Requirements, and written more than 175 articles. Prior to starting Process Impact in 1997, he spent 18 years at Eastman Kodak Company. His responsibilities there included experience as a photographic research scientist, software applications developer, software manager, and software process and quality improvement leader. Karl has led process improvement activities in small application development groups, Kodak's Internet development group, and a division of 500 software engineers developing embedded and host-based digital imaging software products. http://www.processimpact.com.

ABOUT JAMA SOFTWARE

From concept to launch, the Jama product delivery platform helps companies bring complex products to market. By involving every person invested in the organization's success, the Jama platform provides a structured collaboration environment, empowering everyone with instant and comprehensive insight into what they are building and why. Visionary organizations worldwide, including SpaceX, The Department of Defense, VW, Time Warner, GE, United Healthcare and Amazon.com use Jama to accelerate their R&D returns, out-innovate their competition and deliver business value. Jama is one of the fastest-growing enterprise software companies in the United States, having exceeded 100% growth in each of the past four years, during which time both Inc. and Forbes have repeatedly recognized the company as a model of responsible growth and innovation. For more information please visit http://www.jamasoftware.com.