

Automating Requirements Management¹

Karl E. Wiegiers

Process Impact

www.processimpact.com

It's no secret that poorly understood user requirements and uncontrolled scope creep lead to many software project failures. Many software development organizations are improving the methods they use to gather, analyze, document, and manage their requirements. Project teams traditionally document their requirements in a structured software requirements specification (SRS) written in natural language. However, a document-based SRS has some limitations:

- It's difficult to keep current
- It's hard to communicate changes to the affected team members
- It's difficult to store supplementary information about each requirement
- It's hard to define links between functional requirements and corresponding use cases, designs, code, tests, and project tasks.

A commercial requirements management tool that stores requirements and related information in a multi-user database provides a more robust solution. These tools provide functions to manipulate and view the database contents, import and export requirements, define links between requirements, and connect requirements to other software development tools.

In this article, I'll describe several benefits a requirements management tool can provide, identify the basic features to expect from such tools, and briefly review four commercial tools: TBI's Caliber-RM, QSS's DOORS, Rational's RequisitePro, and Integrated Chipware's RTM Workshop (Table 1). These tools won't help you gather the right requirements for your project, and they don't replace a defined process for managing your project's requirements. A tool is not a process in itself, but it supports and enables an established process.

Reasons to Use a Requirements Management Tool

Even if you've done a magnificent job gathering your project's requirements, automated assistance can help you manage them as development progresses. Here are some requirements management tasks these tools can help you perform.

Manage versions and changes. Your project should define a requirements baseline, a specific collection of requirements that a particular release will contain. A history of the changes made to every requirement will explain previous decisions and let you revert to a previous version of a requirement if necessary.

Store requirements attributes. You should store a variety of information, or attributes, about each requirement. Everyone working on the project must be able to view the attributes and selected individuals must be able to update their values. Requirements management tools generate several system-defined attributes, such as date created and version number, and they let you

¹ This paper was originally published in *Software Development*, July 1999. It is reprinted (with modifications) with permission from *Software Development* magazine.

define additional attributes of various data types. Consider defining attributes such as author, person responsible, origin or rationale, release number, status, priority, cost, difficulty, stability, and risk.

Link requirements to other system elements. Tracing individual requirements to other system components helps ensure that your team doesn't inadvertently overlook any requirements during implementation. You can define links between different kinds of requirements and between requirements in different subsystems. When analyzing the impact of a change proposed in a specific requirement, the traceability links reveal the other system elements that the change might affect.

Track Status. Tracking the status of each requirement during development supports overall project status tracking. If a project manager knows that 55% of the requirements allocated to the next release have been implemented and verified, 28% are implemented but not verified, and 17% are not yet fully implemented, then he or she has good insight into project status.

View requirement subsets. You can sort, filter, or query the database to view subsets of the requirements that have specific attribute values.

Control access. You can set access permissions for individuals or groups of users. Web access lets you share requirements information with all team members, even if they are geographically separated.

Communicate with stakeholders. Most requirements management tools let team members discuss requirements issues electronically. E-mail messages can notify the affected individuals when a new discussion entry is made or a requirement is changed.

Commercial Requirements Management Tools

Table 2 compares the major product features for the four requirements management tools I discuss in this article. The tools are still evolving, so their specific features might change with future releases. They all let you define different types of requirements, such as business requirements, use cases, functional requirements, hardware requirements, nonfunctional requirements, and tests. All of the tools integrate with Microsoft Word to some degree. The more expensive tools support a rich variety of import and export file formats.

You can add new requirements directly to the database, or you can select specific text from a document to be treated as a discrete requirement. The tools can also parse an SRS to extract individual requirements. The parsing is imperfect unless you're diligent about using text styles or keywords, such as "shall," when you write the SRS. The tools all support hierarchical numeric requirement labels. Output capabilities include keeping the SRS synchronized with the requirements database and generating a textual requirements document, either in a user-specified format or as a tabular report.

All the tools have robust traceability features, including indicating "suspect requirements" that might be affected if a linked requirement is changed or deleted. They all let you set up user groups and define permissions for selected users or groups to read, write, create, or delete projects, requirements, and attribute values. Most of the tools let you incorporate non-textual objects such as Microsoft Excel worksheets and images into the requirements.

These tools show a trend toward increased integration with other application development tools, such as testing, design modeling, problem tracking, and project management tools. I don't know of any product that seamlessly integrates requirements with every downstream development

work product. However, you can link requirements in RequisitePro to use cases modeled in Rational Rose and to test cases in Rational TeamTest. DOORS lets you trace requirements to individual design elements in Rational Rose. RequisitePro and DOORS can also communicate with Microsoft Project to connect individual requirements to project tasks. Caliber-RM's central communications framework lets you link requirements to use case, class, or process design elements stored in Select Software Tools's Select Enterprise and to test entities stored in Mercury Interactive's TestDirector. You can then access those linked elements directly from the requirement stored in Caliber-RM. Enhanced integration is clearly a strong evolutionary direction for these requirements management products.

Caliber-RM

Caliber-RM is best suited for users who wish to take a database-centric, rather than a document-centric, approach to requirements management. It leads the pack in usability, including many options to let users tailor their preferences. Caliber-RM has a fairly loose integration with Word, but it has highly flexible import capabilities. A flexible Document Factory feature lets you define an SRS template, which is populated with requirement attributes it selects from the database according to defined query criteria to produce an attractive Word document.

Caliber-RM provides a Windows Explorer-like workplace for manipulating the hierarchical requirements tree, with requirement details accessible through a tabbed dialog on the right side of the screen. You can also manage traceability relationships and attributes through a grid display, in which you can easily define traced-to and traced-from links between two requirements of the same or different types. The web component display looks almost exactly like the application, but the displayed requirement attributes are read-only, except for threaded discussion topics. Version 2.0, which should be available by the time you read this, will include a flexible baselining feature. Caliber-RM is a powerful, well-designed, and easy-to-use tool that can handle the requirements for small or large development project.

DOORS

DOORS is a sophisticated product that can manage requirements on large products. It treats individual requirements as objects, but presents them in a visual format that resembles a structured, hierarchical requirements document. A printed report of requirements selected from the database looks like a table-structured SRS. The requirements display also shows attribute values, indicators of links to other requirements, and colored bars that indicate a requirement's change status. DOORS stores the change history of individual objects, modules, and specific edit session, but objects are not given revision numbers.

DOORS includes a C-like scripting language for customizing or developing extensions to the basic product. It also includes direct interfaces to Microsoft Project, Teamwork, and Rational Rose, and you can create interfaces to many other tools through an open API. The link to Word consists primarily of an import capability. DOORS supports many import and export file formats. An integrated change proposal system lets users review and comment on a project or a requirements module. The heterogeneous client/server implementation will be attractive if your organization has a mixed Unix and Windows environment.

I found DOORS less intuitive to use than the other products. In general, the user interface felt inefficient and required considerable "mousing around" from one dialog box to another. For example, to edit a requirement's attributes, you must select the requirement, then select the attribute to edit, and then edit the values in another dialog box. Defining requirement links through the link matrix is clumsy, but DOORS also provides several other link definition

mechanisms. Nonetheless, DOORS is a major contender with many years of marketplace experience behind it.

RequisitePro

RequisitePro is distributed as a component in Rational Suite AnalystStudio. RequisitePro takes a document-centric approach to requirements management, exhibiting the tightest integration with Word of the four tools. In the Word workplace, you can mark selected blocks of text to include in the database as discrete requirements. You can bring up a dialog box to access the requirement details, including its revision history, attributes, traceability, hierarchy, and discussions. The mechanisms it uses for synchronizing the requirements in the database with the contents of the SRS are a bit clumsy. RequisitePro handles the version history of each requirement nicely.

In the Views workplace, you can see the list of requirements in the database and view or modify requirement attributes. The Views workplace is easy to use but not quite as friendly as the Explorer-like hierarchical tree structure that Caliber-RM and RTM Workshop use. It also displays a traceability matrix in which you can easily set and change links. You can archive RequisitePro projects in several configuration management tools or within RequisitePro, thereby managing multiple versions of the project.

I found several usability annoyances with RequisitePro, and the intimate integration with Word leads to some quirks. For example, the user's guide warns not to use Word's cut, copy, or paste functions to move requirements but to use the corresponding RequisitePro functions. Word's undo function can also cause problems. When you mark text as a requirement, it is visually highlighted, and Word bookmarks and hidden text are inserted. If you enter new text immediately following a requirement in the Word document, it takes on most of a requirement's visual highlighting characteristics, even though it hasn't been placed in the requirements database. RTM Workshop has a similar problem. These usability quirks detract a bit from a good tool for introducing requirements management automation into an organization that currently uses structured SRS documents.

RTM Workshop

RTM Workshop is an elegant product that can support the requirements management needs of very large projects. It treats requirements as objects, which belong to classes that correspond to various types of requirements or other project artifacts such as design elements or test cases. Each project defines a class schema in the form of an entity-relationship diagram. Traceability is handled by defining links between objects in two classes (or within the same class), based on the class relationships defined in the schema.

RTM Workshop and RequisitePro integrate with Word similarly, using hidden text and color highlighting to identify text selected to be requirements. In addition, RTM Workshop uses an Explorer-like workplace for manipulating the requirements tree, similar to that in Caliber-RM. RTM Workshop has great flexibility in importing requirements from different types of documents, and it can export requirements in many file formats.

RTM Workshop is designed for effective usability. For example, you can quickly design forms to manipulate the objects in a specific class using a very nice screen painter tool. A simple built-in change request tool links changes to specific requirements. RTM Workshop is a powerful requirements management tool that can meet an organization's needs for a wide range of project sizes.

The Bottom Line

Any of these tools will make your requirements management practices more sophisticated and capable. Select a tool based on the platform, pricing, and requirements paradigm—document-centric or database-centric—that best fits your development environment. Once you've tried a requirements management tool, you'll never go back to plain paper again.

Table 1. Major Requirements Management Tools

Tool	Vendor	Platforms	Document- or Database-Centric
DOORS 4.1	Quality Systems and Software Inc. www.qssinc.com	Windows 95 Windows 98 Windows NT, Solaris, Hewlett Packard-UX, IBM AIX, Digital UNIX	Database-centric
RTM Workshop 5.0	Integrated Chipware www.chipware.com	Windows 95 Windows 98 Windows NT, Unix client; Windows NT or Unix server	Database-centric
RequisitePro 4.0	Rational Software www.rational.com	Windows 95 Windows 98	Document-centric
Caliber-RM 1.1	Technology Builders Inc. www.tbi.com	Windows 95 Windows 98 Windows NT client, Windows NT server	Database-centric

Table 2. Comparison of some requirements management tool features.

Feature	DOORS	RTM Workshop	Caliber-RM	RequisitePro
Parses a source document to load requirements into database	✓	✓	✓	✓
Imports requirements from Word tables into database	✓	✓	✓	
Incorporates non-textual objects such as Excel worksheets and images into database	✓	✓	✓	
Synchronizes textual SRS with database contents		✓		✓
Defines different attributes for different types of requirements and set attribute values for individual requirements	✓	✓	✓	✓
Defines requirement baselines	✓	✓		
Notifies affected project participants by e-mail about requirement changes	✓	✓	✓	
Defines traceability relationships or links between individual requirements and between requirements and other system elements	✓	✓	✓	✓
Tailors usability options	✓	✓	✓	✓
Includes learning aids, such as a tutorial and/or sample projects	✓	✓	✓	✓
Integrates with other tools, such as testing, design, and project management	✓	✓	✓	✓
Defines users and groups and their access privileges	✓	✓	✓	✓
Enables threaded discussions on requirements		✓	✓	✓
Includes web interface for database query, discussion, and perhaps updating requirement attributes	✓	✓	✓	✓
Includes built in change proposal system	✓	✓		
Includes hardcopy user manuals	✓		✓	✓