Making Sense of Requirements Management

If you’re new to requirements management, as many web teams are, this tool can help build successful web applications by organizing and managing your project’s needs.

By Wayne Woodruff

Are you involved in web development, application software development, or any type of product design and development? Is your organization struggling with the concept of developing and maintaining adequate requirements for your webs, products, or projects? Are you being hit with new and complex projects that require frequent updates? How do you maintain these requirements and communicate changes effectively to the team while getting your day job done? RequisitePro, from Rational, Inc., can be an invaluable asset in organizing and managing the requirements management process. It offers (among other things):

- Document organization for all project documents.
- User definable attributes for each document type.
- Dynamic traceability links between documents.
- Open database for managing requirements.
- PVCS Integration.

RequisitePro is tightly coupled with Microsoft Word. Although the tool can work with documents generated with other word processors, you’ll need to save the documents in Word format if you want to edit them while using Requisite. The users of this of this software tend to be application or product development teams, while the purchasers tend to be development of quality assurance managers. The only notable competitor for this product is RTM, which is developed by the Marconi Systems. It was originally a UNIX-based product and has now been ported to Windows.

Creating a project and document types

After invoking RequisitePro, you’ll see the tool palette (figure 1). On your taskbar is an icon for Requisite Views. I discuss that aspect of the tool later on.

![Figure 1: From the Requisite Tool Palette, you manage all the projects, each of the project documents, and the project requirements](image-url)
To start a new project, click on the **Project** button, then select **New**. A project wizard guides you through the process.

The first step is to define a type of document. A document can be of any type you want, although I typically start with the Product Requirements Definition, as all other project documents are derived from this. After defining the type of document you can then import the document (assuming it exists) or you can continue to define the document types and perform the importation later. Repeat this process for each type of document you have. If the documents don’t exist, you can create them within RequisitePro using any Word template. Personally, I don’t put the document under RequisitePro until the project team has reached consensus that at least 80 percent of the requirements are stable. My only reasoning for this is that each time you change a requirement, the revision of that document is incremented and a change history is kept. I would prefer not to keep a long involved history on a requirement that hasn’t evolved. Some folks may want to keep this history for the entire life cycle.

After completing the project wizard, you can open the project from the project tool. When you select the open option, you’re presented with a list of document currently under control of RequisitePro (figure 2). You can open one or more of these documents by selecting them and clicking on **Open**.
Figure 2: The Requisite Project and Document list contains a list of all projects under Requisite control. By selecting a project, the document list for the project will be displayed. Any or all of the documents can be selected for opening in this dialog box.

RequisitePro starts Word and opens the documents. You can modify the document using Word as you would normally. The only difference is that RequisitePro disables many of the Word file commands (only when RequisitePro is open) and forces you to use the RequisitePro equivalent. (This is not a big deal, you quickly get used to it.)

Attributes of requirements

After defining the document types, and before you actually capture requirements, you must decide what attributes you want your requirements to have for each document type. Attributes can be one of these types: numeric (e.g. cost), text field (for comments), or pick list (most attributes are of this type). For example, a Product Requirements Definition may have some or all of these attributes:

**Priority** – What is the priority of this requirement? Is it high, medium, or low?

**Effort** – What is the relative effort of this requirement? Is it going to be very difficult to implement, of average difficulty, or quite simple?

**Scheduled Release** – If you’re doing incremental development, you may want to define what release the requirement will appear in. (Test folks will love this one.)

**Revision** – requisite keeps track of this for you.

**Author** – Who wrote this requirement?

**Owner** – Who is currently responsible for this requirement.

**Status** – I like to use Proposed, Approved, and Postponed. When a requirement is captured, it defaults to proposed. After the team agrees upon it, it’s moved to approved. If we never agree upon it, it gets moved to postponed. The idea is to keep effort from being spent on requirements that may change or be dropped.

**Development Status** – Is it not being worked on, is it being developed, is it being validated, or is it complete?

RequisitePro has default attributes. You can use them, modify them, or create your own. The technique for defining an attribute is to first name it, select the type (text, numeric, or pick list), then create it. Give some serious thought to defining the attributes up front. You can modify or add them later, but it’s easier if you have them correct from the start.
Capturing requirements

Once you’ve defined the attributes for each document type, you need to capture the requirements. RequisitePro, in conjunction with Word, makes this a simple task. In Word, highlight the text you want to be a requirement and click on Create Requirement on the tool palette. RequisitePro generates a requirement number and copies the text, along with all the attributes and their default values, into a database (Access 2.0). The text in the Word document changes color (configurable) and can be either double underlined or SMALL CAPITALIZED (configurable). When you identified the document type, RequisitePro had you define an identifier for your requirements. For instance, if the Product Requirements Definition identifier was ORD, then the first requirement you tag will be PRD1. This identifier and requirement number is displayed in the word document, preceding the requirement text. As you can see in figure 3, PRD3 defines the General ECG Input Requirements.

<table>
<thead>
<tr>
<th>PRD3 General ECG Input Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>The model 750 system shall accept either a low level 5 lead signal or a high level signal from an external monitor for proper triggering.</td>
</tr>
<tr>
<td>PRD4 Signal monitoring shall meet the parts of AAMI standards EC-13 and ECGC specifically listed in this document.</td>
</tr>
</tbody>
</table>

Figure 3: This example of an in-situ requirement shows how Requisite would display a captured requirement within the Word GUI.

Using this technique, you can easily spot the requirement in the Word document. If you need to modify the text of the requirement, place your cursor inside the requirement text and click Change Requirement on the tool palette. A dialog box appears, as illustrated in figure 4. At this point, you can enter some text to describe why the requirement changes and also modify any attributes, such as changing the status from approved to postponed – you did change it, and you do want to ensure that the affected team members are aware, don’t you?

The Requirement dialog lets you see the requirement type, tag, and which document the requirement is located in. The Revision tab allows you to see the revision number with date and time, author, version label, change description, and change history. The Attributes tab permits you to assign new values to each of the attributes.
RequisitePro has the ability to import a Comma Separated Value (CSV) file directly into the database. I didn’t evaluate this feature.

Figure 4: The Modify Requirement dialog box is used to modify aspects of the requirement such as the status of the attribute or the revision number.

Traceability

One key feature of the tool is traceability. If you have a Product Specifications Specification (PRD) as the highest level document that generates a Software Requirements Specification (SRS) and a Software Test Specification (TST), how can you be sure the requirement identified in the PRD was actually designed into the product and that it was tested? If a requirement in the PRD changes, what is the impact on the SRS and TST? If development of testing uncovers a missing or vague requirement, how can you assess the impact on the SRS? Traceability helps you identify missing, incomplete, or incorrect relationships between requirements.

After you have captured all the requirements from the various documents, you can establish traceability relationships. Traceability relationships dynamically track changes to requirements such that if any requirement changes, Requisite automatically notifies you – via the traceability matrix – that linked requirements need to be reviewed. For instance, if you change a requirement in the PRD, Requisite immediately identifies the
impacted requirements in the SRS ad TST. This single feature saves me significant time as I don’t need to review each of the documents in painful detail to determine which, if any, changes need to be made to the other documents.

You can establish traceability relationships in several ways. I’ve found that the database views method is the easiest. If you open the views tool and select New View (figure 5). There are four views to pick from: Attribute and Traceability matrices, and Traced from and Traced to trees. For each of these views, you can filter and sort on all the requirement types and save your customized views in either a personal or project groupings. The Attributes Matrix displays all the attributes for a single document in matrix form (figure 6). This only shows part of the attributes – you can scroll the screen to see all of them. By moving your cursor to the Traced to field and double clicking, the program opens the modify requirement dialog box (figure 4), opened to the traceability tab. You can link this requirement to any other requirement(s) by browsing a list of captured requirements and selecting the desired requirement you want to link to. Repeat this for all your needs.

**Figure 5:** The View dialog allows you to select which type of view you want and what documents the view should be drawn for.
Figure 6: The Attributes Matrix shows you a tabular view of the requirements and attributes for a given requirement type.

Figure 7: The traceability tree shows the requirements of a single document and any traceability relationships.
Figure 8: The Traceability Matrix allows you to show traceability relationships between any two documents.

You might think this is time consuming or a waste of time, but you’ll be convinced otherwise as soon as the first requirement changes and it only takes seconds to identify other requirements you need to modify.

Now that you’ve established all of your traceability relationships, you need to do a little QA on your work. This is where the traceability tree comes in (figure 7). In this example, PRD10 is traced to SRS17,33 and so on. But what about PRD11 through 25? They have no traceability relationships. Did they get forgotten or are they not completely defined? This view will help you quickly identify missing links.

Now that you’ve done some QA on yourself, you can view a traceability matrix (figure 8). This matrix shows the relationship between requirements in any two documents. This example shows the relationships between the PRD and SRS: PRD8 is traced to SRS1 through SRS6. The red line shows what are referred to as “suspect links”. In this example, PRD 8 has changed, which means that SRS1 through SRS6 are suspect and need to be reviewed. After you review those requirements, you can clear those suspect links.

**PVCS Integration**

Requisite integrates seamlessly with Intersolv’s Version manager. Under the Project menu, there’s a selection for PVCS. By selecting it, you get the dialog box in figure 9. This box lets you maintain all Requisite projects under Version Manager control. To place a Requisite project under control of PVCS, click on Add to PVCS, and you’ll see a
dialog that lists all the projects under control of Requisite. Select the project you want to add, and click on OK. Requisite prompts you with a dialog box that lets you select a revision number, version label, and change description. Select OK, and the project is under VM control. All project documents are checked in. Subsequent check-ins and check-outs are on a project basis.

To access a project that’s been checked-in, you simply select the check-out (write) option from the main PVCS dialog box. I always create a Requisite folder in Version manager and add all the Requisite files to the folder. This lets me view the status of the Requisite project inside Version manager. For those who don’t use PVCS, I’m told that an interface to SourceSafe is forthcoming. Requisite also includes an archive function that allows you to keep a snapshot of the project in a different directory, in lieu of PVCS.

Figure 9: The PVCS Version Manager Interface dialog allows you to add a new project to PVCS or manage an existing project under PVCS control.
What’s new in 2.0

With the Pro 2.0 release, Requisite stores requirements in a Microsoft Access database (version 2.0). This is a big step forward for the product; all previous versions used a proprietary database that only allowed access via the Views queries. You can use Access to develop your own queries and reports, but you’ll probably find the querying capability in the Views to be all that you need. Why is such a big deal if you can get all you want from Views? Have you considered reuse of requirements? We all dream of code reuse, why not start with requirements? Don’t limit your thoughts to only product requirements – design and test as well! With Requisite opening it’s database, you can start thinking in this direction.

With the introduction of the open database, Requisite developed a multi-user capability. In previous versions, only a single user could open a Requisite project. Now, up to 255 users can be working in the tool simultaneously, however, only one person can have a document open for writing. Also, in previous versions, you could undo any changes by simple exiting the project without saving the changes. This isn’t the case with a relational database. With Pro 2.0, changes to the database are committed when the document is saved.

With multi-user capability, you need the ability to protect the document for unauthorized or accidental changes. This is accomplished via the security mechanism.

Security is set up on a project basis. A list of users is entered by the Administrator and the users are assigned to groups, based on their needs. Permissions to modify documents are granted on a group basis, much like other multi-user programs I’ve worked with. I’d like to see future versions of Requisite contain an interface to some of the more popular Object Oriented CASE tools (I realize this is like hitting a moving target). Requisite does a great job of managing and controlling requirements, which many of the CASE tools do poorly.

Room for improvement

What don’t I like about the product? Well, as long as all your documentation is in Word–based, you’re in good shape. My biggest concern is that, while I have tremendous visibility in the traceability relationships from the Product Requirements Specification, down through the Software Requirements Specification and the Software Test Plan, I have no visibility down the design/code path. We use a commercially available OO Analysis and Design tool and currently there is no way to get the requirements from the repository into the OO tool in a seamless fashion. Given the creativity and imagination of Requisite, I’m sure they will develop a well-integrated solution to support the major OO tools.
Tech support, maintenance, and training

Requisite, Inc. technical support is quite good (always has been). If TS is really busy, you can leave a message and they’ll call back, usually within a few hours. I’ve sent out a couple of “stumpers” that have taken a few days to solve, but I’ve never had the product corrupt or destroy my data.

Requisite offers maintenance contracts on the software that entitles you to upgrades within your product. There’s a Requisite Baseline product that doesn’t offer the rich feature set that Pro has to offer. When I upgraded to Pro, I paid a fee that gave me full credit to the baseline product I’d purchased a year prior.

They also offer training classes. I haven’t attended, but know several people who have and they highly recommend them. It’s a two-day class on how to manage requirements, followed by a third optional day for training on the tool. You can take the class at their site, or they’ll come to you for a reasonable fee.

Conclusion

You can use this tool to ease into managing requirements whether they be product, project, or web centric. For a small organization, or one new to RM, the traceability tool is worth the price of admission. By simple linking the documents together, you’ll see immediate payback.

I’ve tried to explain the major features of RequisitePro 2.0; it’s quite a complex product and you really need to take a small amount of time to understand the underlying theory of projects and documentation and how they all fit together. Beyond that, it’s fairly easy to use. If you’re new to requirements management, this tool can help you navigate the subtle details, and understand the complexities of managing requirements.

Wayne Woodruff is a principal engineer for the Brooks Instrument Division of the Emerson Electric Company, where he is responsible for software quality assurance and software process improvement. He has a BSEET from Spring Garden College and a MEE from Villanova University. His personal interests include motorcycling, restoration of antique motorcycles, and home improvement projects. wayne@bmwmat.bucks.pa.us