



## Bottom-Up Management

### Understanding the Benefits of RMTrak's Bottom-Up Management Approach

*Overview: Bottom-up management, as opposed to top-down management, allows organizations performing product development to minimize unnecessary rework and keep changes as localized as possible. This ensures the timeliness and success of your organization's product development process.*

RMTrak is RBC Product Development's exciting new requirements management tool for tagging, tracing, allocating, and verifying product requirements. RMTrak is a simple, non-intrusive program that captures, tracks, and manages the project requirements information in your requirements management documents from outside of the document creation process. This lets you use the tools you are already familiar with to format your documents, including Microsoft® Word with its powerful word processing features.

Unlike many requirements management tools, RMTrak uses a bottom-up management approach, rather than a top-down management approach, for structuring requirements documents. This document will help you understand bottom-up management and why it's crucial to successful requirements management.

### Requirements Documents

Requirements documents describe project requirements and should be well structured, complete, coherent, consistent, and correct. Additionally, they should fit their purpose. You should create these documents before you commit design and development resources, each time a change is made to project requirements, and as requirements are implemented/tested.

There are many types of requirements documents, including:

- Customer Requirements Documents
- Regulatory Requirements Documents
- Product Requirements Documents
- Functional Requirements Specifications
- Functional Design Descriptions
- Functional Verification Test Procedures
- Functional Verification Test Reports
- Product Validation Test Procedures
- Product Validation Test Reports

Note: Your organization's project might or might not include product development. Many organizations, and most of the organizations using RMTrak, are familiar with product life-cycles; however, if your organization is not developing a product, *project life-cycle* seems more appropriate. For these reasons, the words *project* and *product* are used interchangeably in this document.

Note that a “functional” document exists for each functional group within a project (i.e. electronics, software, mechanical, hardware, etc.). The exact types of documents and their position within the project life-cycle will be different for each organization.

Requirements documents and are usually ordered so requirements from one document flow down into the next document. With each lower layer, the information can become more detailed and specific. One possible hierarchy is shown below. This particular hierarchy is used in regulated environments, such as medical device design. This hierarchy reflects RMTrak's default document types and their terminal status.

**Tip:** Terminal documents are documents that do not require any additional documentation stemming from their contents.

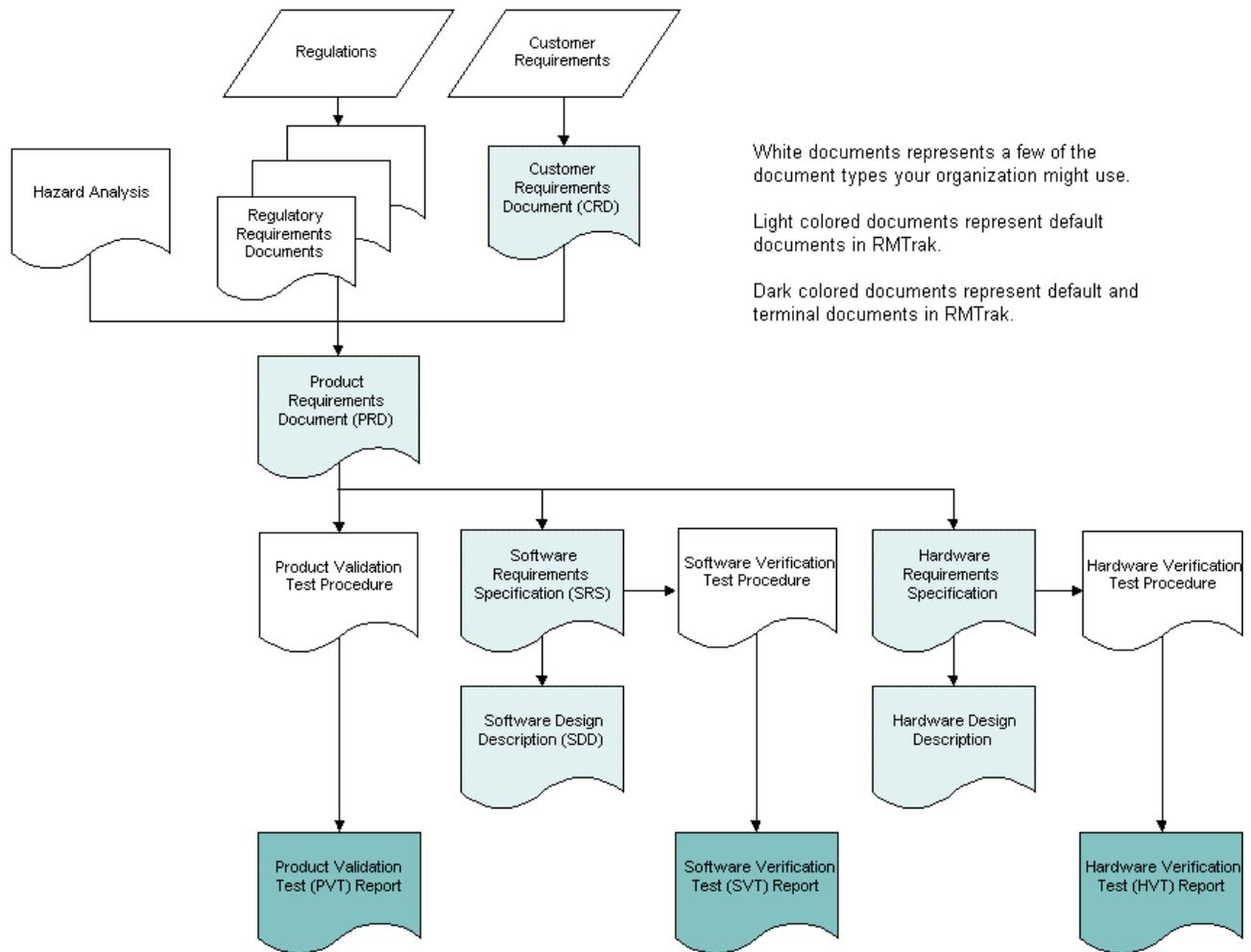


Figure 1: Documentation Hierarchy

Each document's requirements are expanded down through the documents: A customer's initial requirements could each lead to several product requirements, and each of those could lead to several software requirements. Or from a document view, one Customer Requirements Document (CRD) requirement could lead to several Product Requirements Document (PRD) requirements. Each PRD requirement could lead to several Software Procedures Specification (SRS) requirements, and so on. In this way, a project that began with a few requirements could eventually have several thousand. If extra requirements are added during a project, the changes need to filter down through every document, as each requirement near the top may affect several requirements in the document after it.

## Requirements Management Tools and Document Tags

To help keep track of requirements, many organizations use a requirements management tool like RMTrak. These tools keep track of project requirements and how they relate to each other.

To use the tool, document authors assign tags to the requirements. These tags allow the requirements management tool to track the requirements and their relationship to requirements in other documents. Authors can tag Microsoft® Word documents manually or insert the tags using RMTrak's macro features.

In the below example, **PRD: 3** is a tag showing that the requirement is the Number 3 requirement in the Product Requirements Document (PRD). The number of the tag (the 3 in **PRD: 3**) is an arbitrary number and doesn't have significance other than being a unique identifier.

**PRD: 3** The enclosure must withstand being dropped from a height of six feet without any observable damage.¶

When you import documents into RMTrak, RMTrak recognizes the documents' requirements by their tags and formatting. For example, RMTrak will recognize tags if they are formatted to be both bold and red or if they are formatted in a Microsoft® Word style. RMTrak recognizes the end of requirements by their ending paragraph tag (¶) or by the character you specify in RMTrak's settings.

## Top-Down Management

Because each requirement affects document requirements that come after it, most product development companies take a top-down approach to requirements management. As team members write documents for their functional area (the software team writes the Software Requirements Specification, the hardware team writes the Hardware Requirements Specification, etc.), they focus on the requirements in their documents and on the requirements in the documents below them in the documentation hierarchy.

If the team's requirements management tool takes a top-down approach, the team will need to format the tags in each of its documents to reflect the requirements in the documents that will come after it. This approach lets the tool recognize which requirements will affect other, future requirements.

In the following example, a requirement in the Product Requirements Document (PRD) affects three other requirement tags (formatted in brackets) in the Hardware Requirements Specification (HRS).

**PRD : 3** The enclosure must withstand being dropped from a height of six feet without any observable damage.  
**[HRS : 4 , HRS : 6 , HRS : 32] ¶**

The initial tag is arbitrary and doesn't have significance other than being a unique identifier. The end tags (here "child tags") are extremely important and must exactly match the tags given to the document requirements that came before them. Unfortunately, this approach is costly and timely.

### **An Example of Top-Down Management's Inherent Problems**

The Hardware Team, for instance, might re-organize and re-number the requirements in its Hardware Requirements Specification. The Product Requirements Document now has incorrect child tags that point to the wrong tags in the Hardware Requirements Specification. To fix this problem, the Product Requirements Document needs to be changed so it reflects the correct Hardware Requirements Specification tags.

Unfortunately, because the Product Requirements Document is an important document that often must go out to the Client each and every time a change occurs, the Client is bothered with unnecessary hindrances and additional costs for the time of meetings, change revisions, etc.—all to edit a few numbers. Additionally, because the Product Requirements Document was originally created many months ago, at the beginning of the project, revisiting it means going backwards in the project.

### **Bottom-Up Management**

The initial documents of a project are typically the most important, as they provide overviews, while the subsequent documents tend to be more detail-focused. Initial documents should be created at the beginning of a project and not be repeatedly edited. Once the top-level requirements have been finalized, they should only be edited if additional requirements are added to the project or if existing requirements change. In this case, a comprehensive cost and/or impact analysis would be performed prior to implementing the changes.

With bottom-up management, author(s) format a document's tags to reflect requirements in the documents above it. In the following example, a requirement in the Hardware Requirements Specification (HRS) originates from a requirement (formatted in brackets) in the Product Requirement Document (PRD).

**HRS : 4** The enclosure must withstand being dropped from a height of six feet without any observable damage.  
**[PRD : 3] ¶**

The initial tag is arbitrary and doesn't have significance other than being a unique identifier. The end tags (here "parent tags") are extremely important and must exactly match the tags within their parent documents. This helps RMTrak track the lineage of each requirement and where it originated in the project.

## **An Example of Bottom-Up Management's Effectiveness**

If the hardware team were to re-organize and re-number the requirements in its Hardware Requirements Specification, higher-level requirements documentation would not be affected. This is important because it means the document before it, which was considered finished, won't need to be re-opened and updated. It can remain finished.

Unfortunately, this means that any documents that come after the Hardware Requirements Specification will need to reflect its new tags. Fortunately, in this scenario, the other documents haven't been started yet. Thus, there will be no impact.

A bottom-up management approach ensures that changes to a document only affect subsequent documents, which probably haven't been created yet, and that finished documents remain finished and needn't be changed for the differences made in later documents.

In this way, bottom-up management allows product development processes to:

- Minimize unnecessary rework
- Keep changes as localized as possible

## **RMTrak and Bottom-Up Management**

RMTrak uses a bottom-up management approach. The tags you place around requirements will include "parent tags" to ensure current documents always points back to the correct requirements in the project's initial documents. This saves your organization time and money, which in turn helps your projects succeed.

## **Additional Information**

RBC Product Development is dedicated to helping you succeed. If you need additional information about bottom-up management, please don't hesitate to contact us or visit our RMTrak Web site.

RMTrak Web site: <http://www.RMTrak.com>

General inquiries: [RMTrak@rbccorp.com](mailto:RMTrak@rbccorp.com)

Technical Support: [RMTrak.Support@rbccorp.com](mailto:RMTrak.Support@rbccorp.com)

---

### **Bottom-Up Management: Understanding the Benefits of RMTrak's Bottom-Up Management Approach**

RMTrak is a trademark of RBC Product Development in the United States and/or other countries. Microsoft is a registered trademark of Microsoft Corporation in the United States and/or other countries.

Copyright © 2002 RBC Product Development. 13715 West 109th Street, Lenexa, KS 66215.  
All rights reserved.