

# Chapter 16: Managing System Requirements

System requirements management is an activity that spans throughout the IT systems life cycle. It relates to the changes and implementation of requirements during the systems development and maintenance. How requirements are initially gathered and stored often reveals the level of an IT organizations engineering discipline. Those that provide teams with an automated requirements environment will better support change control efforts, gain testing efficiencies and potentially reduce their future maintenance burden (Gartner Group, 1998). This chapter presents the process and tools of system requirements management.

## Requirements Management Process

The requirements management process defines and maintains new and changed requirements throughout the systems life cycle. The IT project manager sets up a domain expertise analyst team to keep current, to track, and to trace the requirements throughout all phases of the IT systems development and maintenance. According to James Martin, over 50% of all project defects can be traced to the requirements management process (Figure 16–1).

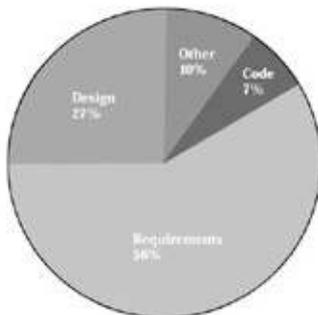


Figure 16–1: Identification of defects in system requirements

Conventional methods of requirements management capturing and communicating requirements in a document, translating requirements from one source to another, and inadequately communicating changes to requirements are not effective in the current fast-paced, distributed development environment. Often practitioners are challenged to come up with what if questions that are related to the problem that they are trying to solve. In analyzing the requirements management problem, the following what if questions often come into play.

- What if the requirement specifications are treated as a group of integrated, reusable objects instead of a static document?
- What if requirements can be captured at their source instead of being gathered and translated from one source to another?
- What if requirements and their associated attributes and representations are stored in a central repository where they can be communicated to and collaborated on by the entire organization?
- What if all users, customers, and stakeholders are electronically notified when a requirement changes?

Months or years before project completion, the ability to effectively manage requirements determines how, when, and how expensively completion will take place. Before processing a requirement, the project manager must develop a schema for the requirements management database.

Attempting to manage requirements through a document can be time consuming based on the volatility of change, and in some cases attempting to keep the document up to date can be an impossible task. Instead of collecting requirements into documents, what would happen if each requirement were treated as an object

## Requirements Management Tools

with specific attributes and behaviors? Treating requirements as objects gives them the following:

- **Visibility.** Requirements can be viewed, sorted, and filtered on an individual basis as opposed to being buried in a document. This approach provides a mechanism for clearer elicitation, analysis, and communication of requirements.
- **Reusability.** Requirements can be reused from one project to the next. This approach provides for versioning of requirements within a software project that has multiple releases.
- **Testability and traceability.** These attributes are defined in Chapter 15.
- **Maintainability, safety, and security.** Each requirement can have its own change history and level of security. This provides an individual record of changes, identification of who made the change, and the reason for the change, rather than forcing users, customers, and stakeholders to review a specification document annotated with change bars.
- **Characterization.** Several pieces of data of various types can be stored for each requirement. In addition to a text-based description, each requirement can be linked to other objects, such as process models, use cases, spreadsheets, documents, and test cases.

Although organizations attempt to do a good job of requirements management, Easterbrook encountered several types of requirement errors (Figure 16–2). Each of these errors is a different issue to organizations developing a system based on the incorrect requirements. The involvement of the users, customers, and stakeholders throughout the process can help mitigate these types of errors.



Figure 16–2: Types of system requirement errors

## Requirements Management Tools

Requirements management tools assist in the management of requirements and have become an important aspect of system engineering and design. Considering the size and complexity of development efforts, the use of requirements management tools has become essential. The tools that requirement managers use for automating the requirements engineering process have reduced the drudgery in maintaining a projects requirement set and added the benefit of significant error reduction. Tools also provide capabilities far beyond those obtained from text-based maintenance and processing of requirements. Requirements management tools are sophisticated and complex because the nature of the material for which they are responsible is finely detailed, time-sensitive, and highly internally dependent, and they can be continuously changing.

Many requirements management tools are available, including simple word processors, spreadsheets, relational databases, and tools designed specifically for the management of requirements, such as DOORS (Quality Systems & Software, Mt. Arlington, NJ) and RTM Requirements Traceability Management (Integrated Chipware, Inc., Reston, VA).

IT organizations that are practicing requirements management can store their requirements in Microsoft Word documents or Excel spreadsheets. Some organizations use departmental databases that may include Microsoft Access or Lotus Notes. Management of requirements in a centralized repository greatly improves the

## Selection Criteria of a Good Requirements Management Tool

requirements management process. It initiates the standardization and sharing of requirements across all projects, allows requirements to be easily traced from inception through deployment and maintenance, and provides a collaboration mechanism in which threaded conversations among customers can be associated with individual requirements. Most importantly, it provides more effective and efficient impact analysis, trade-off analysis, and risk analysis of requirements.

The selection of a tool is only part of the equation. A thorough understanding of the tools capabilities and the management processes that will use the tool is necessary. The tool should not be plugged into the management processes with any thought as to the influence on the tools capabilities. Adjustments may be needed in the management processes and employment of the tool to bring about an efficient requirements management process.

The established requirements management process should have each group responsible for the data in its domain without consideration for how one data set relates to another. As a result, the requirements management tool is set up to have a class for each of the organizational elements without regard to the following:

- How the requirements are being managed in each class
- Whether the requirements are relating cleanly to requirements or tests of other classes

When requirements are stored in documents, the process of notifying the affected users, customers, and stakeholders presents two challenges:

1. The individuals making changes may not know everyone who is affected by the change.
2. Even when they do know who should be notified, they cannot always ensure that notification is accomplished in a timely manner.

Using an electronic form of notification, such as e-mail, to distribute changes ensures that notification is accomplished in a timely manner. The electronic change notification should inform the customer of who made the change, what was changed, why it was changed, and when it was changed. This approach ensures that the users, customers, and stakeholders promptly learn about all changes that affect them. When the customer combines this approach with requirement traceability, he or she can easily accomplish impact analysis of the change.

## Selection Criteria of a Good Requirements Management Tool

- The technology selected must support an organizations best practices for system development. Every organization handles requirements management differently. Based on the scope and complexity of a project, the time and activities allocated to requirements management can span a wide spectrum.
- Since the project manager is ultimately responsible for all requirements, the technology must not add to an already daunting list of tasks and activities. The technology must allow the project manager to effectively control the scope of the project and easily assess the influence of changes to requirements and the project schedule.
- The technology must be easy to use. It must be easy for a variety of customers to learn and incorporate it into their routine system development activities. If the project involves business managers and end-users in the requirement management process, the tool must fit into a familiar context. If it is to be used by developers, it must be robust enough to support daily usage.
- The technology must simplify generation and distribution of specification documents. Requirements specification documents will always be needed to support the new process. Every organization has unique formats for their specification documents, and the technology should easily support generation

## TBI Caliber–RM Tool

of any style of document. Distribution of documents in electronic form via the Internet is essential to the way distributed development teams work.

Based on these criteria, four proven technologies can effectively and efficiently support the reengineered requirements management process:

1. An object–oriented database that is easily customizable and provides a central and secure repository for requirements definition, analysis, and traceability
2. An Internet–based tool that provides visibility, collaboration, and immediate electronic change notification
3. An easy–to–use graphic user interface based on Microsoft Windows, Word, and Windows Explorer
4. A customizable document and report generation and electronic distribution tool

A requirements management tool that tightly integrates these four technologies provides the project team with an efficient and effective way of managing all requirements from inception through deployment.

## TBI Caliber–RM Tool

TBI Caliber–RM is a collaborative, Internet–based requirements management tool that facilitates requirement definition and management throughout the IT systems development cycle. Providing a centralized requirement repository and automatic change notification, the tool enables collaboration and communication among project teams, assisting them in identifying and eliminating requirement errors earlier in the application life cycle. Caliber–RM also allows team members to compare project baselines to easily manage scope creep and identify factors that may affect schedules and budgets. The tool supports reusable requirements and ensures that project teams can build from previous experience and applications, enabling more rapid development and better use of resources. The organizations can instill discipline in their development cycle through a structured requirements management process, minimizing the cost of reworks due to requirement errors, focusing team members on the project scope, and decreasing the likelihood of project failures and overruns. The following are the main features of the TBI Caliber–RM tool:

- Requirements–driven development and testing
- Clear communication
- Shared requirements
- Internet–based nature
- Traceability throughout the life cycle
- Change management of requirements
- External references
- Automatic requirement import
- Comprehensive reporting
- Integration of the life cycle
- Flexibility to support processes

## Requirements–Driven Development and Testing

Various studies have shown that roughly half of all application errors can be traced back to requirement errors and deficiencies. Thorough documentation and proper management of requirements are the keys to development of quality applications. By allowing project teams to define and document requirement data including user–defined attributes, priority, status, acceptance criteria, and traceability Caliber–RM enables earlier detection and correction of missing, contradictory, or inadequately defined requirements. The tool even recognizes terms used in multiple requirements and allows users to link those terms to a standard definition,

## Clear Communication

promoting their correct usage throughout the project.

With a more complete and accurate set of requirements, project teams are able to focus on requirements within the project scope. Using Caliber–RM traceability, project managers are able to easily identify and manage scope creep, keeping the development effort on track and within budget. Through Caliber–RM test planning and management integration, quality assurance (QA) teams can quickly identify the influence that requirement changes may have on testing, enabling them to review and rerun affected tests to ensure that the application will meet end–user needs.

## Clear Communication

Development of a quality application requires a considerable amount of communication among the project team, which may include business analysts, managers, development and QA teams, and customers. Providing a centralized repository for documented discussions about requirements, Caliber–RM can make communication easier and more effective for the entire project team. Team members can enter feedback on projects and requirements in Caliber–RM threaded group discussion, allowing others to review comments for better requirement revision and prioritization.

Caliber–RM also allows project managers to assign individuals to each requirement, allowing them to better manage and understand team member responsibilities. When a requirement is added, modified, and deleted, responsible individuals are automatically notified by e–mail, ensuring that everyone is aware of the current state of requirements at all times. Users also may register interest in requirements, allowing them to receive change notifications even if they are not assigned to a requirement.

## Shared Requirements

Although most application requirements are unique, some requirements may be duplicated within and across projects. To minimize the effort necessary to manage such requirements in multiple locations and reduce the possibility of errors, Caliber–RM allows project teams to share requirements within and across projects. Shared requirements are listed within the requirement tree, and their descriptions are available as read–only. When shared requirements must be modified, the project team need only update them in one location, and the new data are displayed in all linked instances.

## Internet–Based Nature

As project teams grow to include more than just developers and testers, they often expand geographically as well. Caliber–RM is designed to work efficiently over TCP/IP connections and modems. Transmitting only the information that each user needs for faster communication, Caliber–RM can bring distributed project teams closer together. The tool also offers a web access component that allows team members to view and comment on requirements through a Java–enabled browser such as Netscape or Microsoft Internet Explorer. This component provides remote access for individuals only needing review and minor updates, or comment capabilities.

The Caliber–RM user interface is designed for ease of use, with a requirement tree on the left similar to that of Windows Explorer and tabs containing requirement data on the right (Figure 16–3).

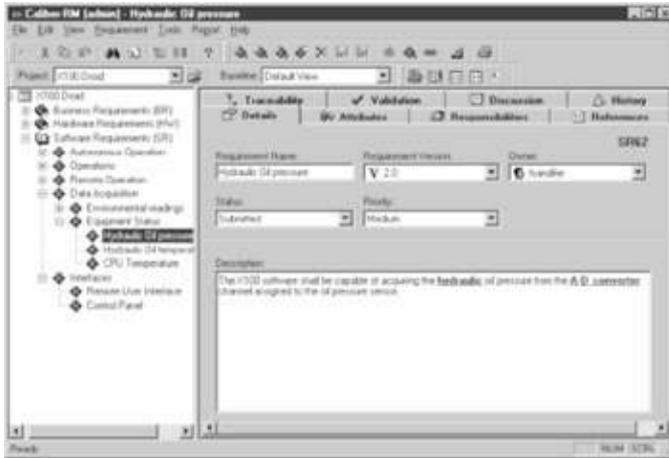


Figure 16–3: Caliber–RM user interface screen

## Traceability Throughout the Life Cycle

Application requirements are rarely isolated fragments of data; rather they work together to form a cohesive unit of information. Understanding the relationships between requirements is critical to understanding the project as a whole. Caliber–RM allows traceability of requirements throughout the application life cycle, managing links between requirements to allow impact analysis and promote a better understanding of the project.

With Caliber–RM, users can see how each requirement relates to other requirements within and across projects. Through integration with third–party tools such as object modeling and test planning and management users also can follow a requirements progress through development and testing by linking requirements to use cases and tests. Using the traceability matrix, users can view all requirement relationships at once, including those between requirement and use cases or tests. When a requirement changes, Caliber–RM automatically marks all associated links as suspect to alert users of possible inconsistencies caused by the change.

Caliber–RM allows users to trace requirements to other requirements, develop entities such as use cases and business processes, and test entities such as tests and test sets (Figure 16–4). When a requirement is modified, Caliber–RM marks all associated links as suspect.

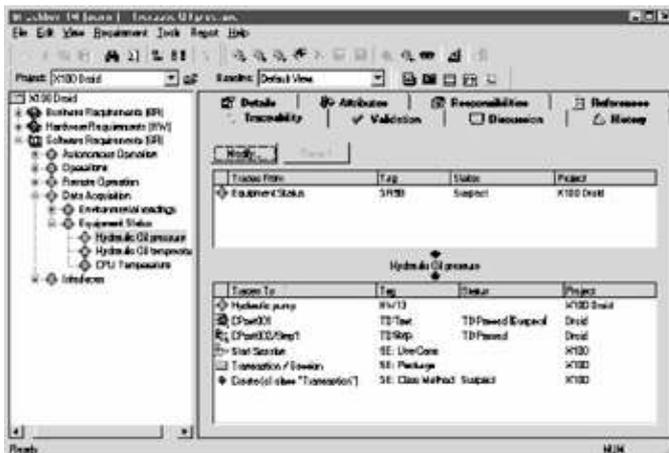


Figure 16–4: Caliber–RM requirement traceability screen

## Change Management of Requirements

When developing an application, the project manager must understand how the requirements evolve throughout the development life cycle. The manager must record changes and the reasons for them to understand how and why the application has deviated from the original specifications. Caliber–RM maintains a full change history for each project and requirement, including the changes that were made, when and why they were made, and who made them. When a requirement is modified, Caliber–RM automatically increments the version number. Users can view any requirement version simply by selecting the version in a drop–down list. Users can also compare versions to better understand how the requirement has evolved.

In addition to having multiple requirement versions, project teams can use baselines to label a subset of requirements at specific versions or to capture a snapshot of the project at a point in time (Figure 16–5). Users then can view a previous baseline to see what changes have been made to the project. Users even can compare baselines to easily identify major changes and scope creep that could affect deadlines and budgets.

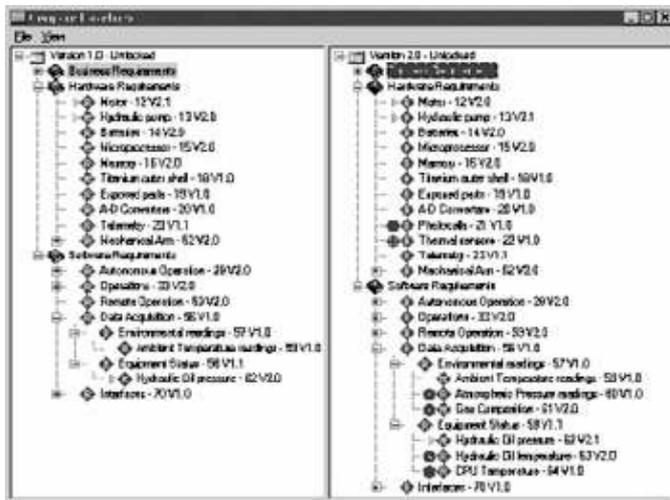


Figure 16–5: Caliber–RM project requirement baseline

Caliber–RM allows project teams to create project baselines at any point in time. Users then can compare baselines to more easily identify scope creep and understand the evolution of the project.

## External References

Often, requirements need supporting information such as screen prototypes, flow diagrams, and spreadsheets to help the project team understand them completely. To make this information accessible to team members, Caliber–RM allows multiple external document references to be created for each requirement. These references can be any type of file, including graphics, spreadsheets, Word documents, and hypertext markup language (HTML). In addition, users can create uniform resource locators (URLs) and text references. Once references are created, users can double–click and open the file. Caliber–RM is closely integrated with Microsoft Word and Excel and allows users to link to text or cells within documents to specify exactly what information relates to the requirement.

## Automatic Requirement Import

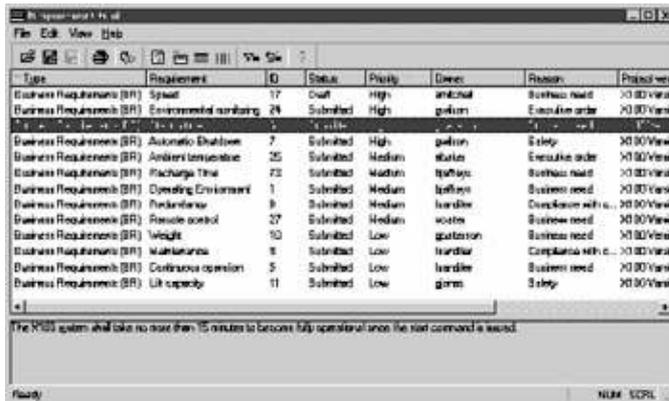
Although an automated requirements management system provides a wealth of power and flexibility for management of requirements, project teams sometimes must author requirements outside of the system, or they may receive the requirements from an outside source. For example, requirements established in a meeting can be compiled in a spreadsheet, or a customer can deliver a specifications document in Word. To

## Comprehensive Reporting

enable teams to leverage existing documentation and incorporate requirements from external sources, Caliber–RM provides an automatic import capability. Using Caliber–RM Import from Word Wizard, users can specify type styles, keywords, and delimiters. The Wizard then parses the specified document and imports all requirements that match the criteria. Users can rearrange, modify, or delete imported requirements before placing them in Caliber–RM to be managed.

## Comprehensive Reporting

Because each organizations development process is unique, Caliber–RM is designed for maximum flexibility. To support project teams that must produce hard copy reports, Caliber–RM provides customizable and standard reporting capabilities. The Caliber Document Factory™ allows users to create any type of customized specifications document using Word templates. These templates can use a filter and sort criteria to print a specific subset of requirements, or the templates may include all requirements. Furthermore, the Caliber–RM standard reports, requirement grid, and traceability matrix offer additional reporting options (Figure 16–6). The Caliber Document Factory allows project teams to create customized specifications documents using Word templates. Users then can generate documents as needed and use them as reports or to create project plans for development.



Type	Requirement	ID	Status	Priority	Level	Reason	Project view
Business Requirements (BR)	Speed	17	Draft	High	initial	Business need	> 00/View
Business Requirements (BR)	Environmental monitoring	24	Submitted	High	gation	Executive order	> 00/View
Business Requirements (BR)	Automatic Shutdown	7	Submitted	High	gation	Safety	> 00/View
Business Requirements (BR)	Antidisturbance	25	Submitted	Medium	stake	Executive order	> 00/View
Business Requirements (BR)	Recharge Time	23	Submitted	Medium	gation	Business need	> 00/View
Business Requirements (BR)	Operating Environment	1	Submitted	Medium	gation	Business need	> 00/View
Business Requirements (BR)	Redundancy	9	Submitted	Medium	gation	Compliance with r...	> 00/View
Business Requirements (BR)	Residue control	27	Submitted	Medium	gation	Business need	> 00/View
Business Requirements (BR)	Weight	10	Submitted	Low	gation	Business need	> 00/View
Business Requirements (BR)	Maintenance	8	Submitted	Low	gation	Compliance with r...	> 00/View
Business Requirements (BR)	Continuous operation	5	Submitted	Low	gation	Business need	> 00/View
Business Requirements (BR)	Lift capacity	11	Submitted	Low	gation	Safety	> 00/View

Figure 16–6: Caliber–RM requirement grid screen

## Integration of the Life Cycle

In any application development project, the true power of requirements lies in how they are used to facilitate development and testing. Requirements that are interactively linked with development and testing objects such as business processes, use cases, and test cases are more likely to be developed and tested correctly.

Caliber–RM is supported by a framework that provides the means to link Caliber–RM with custom and third–party applications including test planning and management, object modeling, problem/change request tracking, project management, and more enabling project teams to manage applications more effectively throughout the life cycle. Caliber–RM currently supports integration with best–of–class object modeling and test planning and management tools.

Through Caliber–RM object modeling integration, users can trace requirements to business processes, use cases, classes, methods, and packages. When a requirement changes, Caliber–RM marks the links as suspect to allow users to see which development entities are affected by the change. Through the test planning and management integration, team members can trace requirements to test sets, tests, and test steps. Caliber–RM automatically displays the status of each test entity and the status of the link. Through the traceability matrix, users can easily identify requirements not linked to development or testing entities to ensure that all requirements are correctly developed and thoroughly tested. Using the Caliber–RM Test Wizard, team members can generate tests automatically, including the verification information stored with each requirement to form the basis for the test. Caliber–RM automatically generates traceability links to the created tests.

## Flexibility to Support Processes

Because each application is unique, an automated requirements management system must be flexible enough to adapt to any needs. Caliber–RM support for multiple requirement types, user–defined attributes (UDAs), and easy–to–use customization features make it flexible enough to support a variety of processes and applications. Using Caliber–RM reusable requirement types, project teams can organize requirements in a manner that best supports their development process. Caliber–RM UDAs allow teams to specify what data should be captured for each of those requirement types. Caliber–RM also allows users to filter and sort requirements based on their attributes through an updatable requirement grid, enabling them to examine and easily modify subsets of requirements. Users can even create custom tabs for each requirement type or hide unused tabs. The result is an interface that supports the needs of the project team, rather than forcing the project team to be controlled by the interface.

The Caliber–RM requirement grid allows users to filter and sort requirements based on their attributes, enabling them to focus on a subset of requirements (see Figure 16–6). Users can update data for each requirement individually or for a group of requirements at once.

## Applications Support

Each type of application contains its own set of rules and constraints. Caliber–RM is designed to support a variety of processes and applications, allowing project teams to customize its interface to meet the needs of their environment. In addition to supporting client/server and Internet applications, Caliber–RM also supports e–commerce, enterprise resource planning (ERP), and supply chain management applications.

## E–Commerce

E–commerce applications are rapidly changing the way that companies conduct business, with the potential of a worldwide market driving tighter deadlines and a greater need for quality. E–commerce applications must be secure enough to prevent data theft, reliable enough for continuous operation, flexible enough to handle rapidly changing content, and accurate enough to ensure precise data transfer. However, too often team members see formal requirements management processes as too restrictive and time–consuming for such a rapid development environment. However, because roughly half of all application errors can be attributed to requirement errors, a requirements management process is critical to delivering the level of quality that customers demand.

Caliber–RM can help project teams develop and implement e–commerce applications by providing them with the means to detect and eliminate requirement errors and deficiencies earlier in the development life cycle and communicate rapidly changing requirements efficiently. Team members then can concentrate on the project without having to worry about requirement errors that could lead to last–minute changes and instability. The result is a higher–quality application that supports the goals of conducting business in a worldwide market.

## Enterprise Resource Planning

Many organizations use ERP, or packaged applications, to support their business rather than developing their own systems. Rapid growth, mergers, and globalization make it imperative that those separate systems work together, with a minimum of effort required for integration. With its centralized requirement repository and support for distributed project teams, Caliber–RM can help organizations maintain application configuration and customization requirements in one location yet allow access by team members in many locations. Caliber–RMs user–defined attributes allow project teams to report on data specific to each module and help ensure more correct interpretation of each requirement. Caliber–RM is an efficient and effective system for

## Supply Chain Management

managing the scope of ERP systems.

To facilitate development of interfaces between modules and existing customer applications, Caliber–RM provides cross–project traceability of requirements, between requirements, and between development and test entities. The Caliber–RM group discussion feature provides an electronic bulletin board for team members and end–users to make change requests for specific application requirements. Project managers then can review the requests to determine the effort, time, and cost required to implement the changes before approving them. Caliber–RM provides automated impact analysis based on traceability of requirements from inception through deployment. As future releases of ERP modules are implemented, the influence of changes to existing modules can be viewed in a traceability matrix before implementation.

## Supply Chain Management

As businesses expand, product lines move into new markets and their supply chains become increasingly complex. New suppliers, additional manufacturing needs, a larger inventory to store and manage in the distribution centers, and time–to–market pressures can turn a simple supply chain into a formidable challenge practically overnight. To remain competitive, improve customer service, and enhance financial results, organizations must develop and maintain a unified supply chain, integrating all of the people, processes, and systems that are involved from beginning to end.

Caliber–RM can help organizations record and control their dynamic supply chain requirements from the overall vision to the individual applications and processes. By providing a single location for documentation and management of requirements, Caliber–RM enables project teams to collaborate on each project, minimizing tangents from the original vision and focusing team members on the established requirements. Caliber–RM's cross–project traceability enables project teams to understand how different parts of the system relate to one another, facilitating a better understanding of the system as a whole. With Caliber–RM, project teams are able to meet supply chain challenges head–on, allowing them to develop and maintain a cohesive system under dynamic conditions.