

White paper

October 2008



Rational software

Ten steps to better requirements management.

Dominic Tavassoli, IBM

Contents

- 3** *Introduction*
- 3** *Defining a good requirement*
- 4** *Ten steps to better requirements management*
- 10** *Conclusion*

Introduction

Requirements definition and management is recognized as a necessary step for the successful delivery of systems and software projects; the discipline is also required by standards, regulations and quality improvement initiatives like Capability Maturity Model Integration (CMMI).

Creating and managing requirements is a challenge for IT, systems and product development projects or indeed for any activity where you have to manage a contractual relationship. Organizations need to effectively define and manage requirements to help ensure they are meeting customer needs, while addressing compliance and staying on schedule and within budget.

The impact of a poorly expressed requirement can be devastating; it can have a domino effect that leads to time-consuming rework, inadequate deliveries and budget overruns. Even worse, a poor requirement can bring a business out of compliance or even cause injury or death.

Requirements definition and management is an activity that has the potential to deliver a high, fast ROI. This white paper explains the characteristics of a good requirement and presents ten steps to better requirements management.

Defining a good requirement

Because requirements are the foundation of any development project, teams need to understand the attributes of a good requirement. The best requirements are:

- *Correct (technically and legally possible).*
- *Complete (express a whole idea or statement).*
- *Clear (unambiguous and not confusing).*
- *Consistent (not in conflict with other requirements).*
- *Verifiable (it can be determined that the system meets the requirement).*
- *Traceable (uniquely identified and tracked).*
- *Feasible (can be accomplished within cost and schedule).*
- *Modular (can be changed without excessive impact).*
- *Design independent (do not pose specific solutions on design).*

Each requirement must first form a complete sentence, containing a subject and a predicate. These sentences must consistently use the verb “shall,” “will” or “must” to show the requirement’s mandatory nature, and “should” or “may” to show that the requirement is optional. The whole requirement specifies a desired end goal or result and contains a success criterion or other measurable indication of the quality.

Ten steps to better requirements management

Once these basic but necessary rules are applied, there are ten steps that help organizations better define and manage requirements.

Step 1: Structure requirements.

Duplicate requirements can cause work to be performed twice, lead to conflicts, and eventually double your maintenance cost. Omitted requirements may lead to missing functionality or cause shortcomings (see below, “Constraints”). Requirements should be structured to enhance understanding while avoiding duplication and omission. Traceability to higher- and lower-level requirements enables teams to assess coverage.

Structuring requirements is the first step in taking control and improving the quality of requirements.

Step 2: Manage and link customer needs, requirements and contracts.

Organizations typically collect the customer's needs, captured "as is." These needs undergo an internal translation to requirements in a format that meets the "good" requirements characteristics described above. They may also be made more generic and less customer-specific (so the system can meet multiple customer needs). There is also often a stable contractual agreement, a legally binding third document. Organizations need to capture these levels of user requirements, maintaining intelligent traceability and change impact analysis between them.

Specifications and contractual documents should be generated from the requirements repository; this central location should also maintain links to outside elements (e.g., customer documents, e-mails and contracts).

By managing the multiple representations of customer needs, organizations have better control over contractual agreements and increase the chance of project success.

Step 3: Manage constraints.

Requirements must not only describe functional behavior. Nonfunctional requirements, also called constraints, can be critical for compliance and regulations and can add quality to the system. Typical nonfunctional requirements can specify:

- *Performance.*
- *Interface.*
- *Security.*
- *Safety.*
- *Reliability.*
- *Availability.*
- *Maintainability.*

Writing better requirements includes providing coverage for constraints since shortcomings in these areas (e.g., performance, reliability and ease of use) generally cannot be reengineered back into the system once developed. By ensuring that they take into account all types of constraints relevant to their industry, organizations greatly increase their projects' chances of success.

Step 4: Visualize requirements.

Most requirements analysts find augmenting textual requirements with modeling helpful, whether this means drawing pictures on a whiteboard, utilizing presentation tools such as Microsoft® PowerPoint or simply creating a mental model. These representations should be managed alongside the requirements to help ensure consistency, traceability and change control. Visual requirements modeling provides a simple and powerful way to communicate with, and elicit requirements from, customers and end users. It also helps clarify requirements and create a common understanding between all development team members and stakeholders.

Although models and images should not replace clear, unambiguous textual requirements, by empowering visual requirements, organizations increase communication and collaboration across all stakeholders.

Step 5: Test requirements.

An efficient way to better manage requirements is to ensure they are clearly mapped to test cases. Making sure each requirement is clearly verifiable from the start not only helps prepare later phases of the project, but it also puts the writer in the correct state of mind. Note that this is true for the nominal functional mode (making sure the system or software does what it's supposed to do). Requirements and their associated tests must also indicate what the system should not do, and what happens at the limits (degraded mode).

This rule also applies to constraints (nonfunctional requirements): Indicating how they shall be tested is a good way to write better requirements. For instance, how would we test the requirement “The software must be highly usable”? A better requirement would be, “An untrained user will be able to generate a report in less than three minutes,” for instance.

Organizations that ensure their requirements are clearly testable, early on in the process, can improve project success rates and enhance quality.

Step 6: Bridge the chasm between business and development.

In many cases, the route to better requirements management is to have fewer requirements. Projects cannot always offer the luxury of implementing all customer requests, marketing ideas and business suggestions when they also have to meet budget and deadline objectives. Rather than trying to manage every requirement, project and product managers must be able to make decisions on those requirements that bring the most value to the customer and help the business improve innovation. This can be achieved by combining value and priority information from stakeholders and defining the right combination of requirements.

By creating and maintaining this link between engineering requirements and business and customer needs, senior management can help ensure that resources are spent efficiently. Development and implementation can similarly align technical decisions with the organization's strategy.

Step 7: Control change to requirements.

Requirements are subject to continual change. As a project progresses, organizations need to remain agile, adapt to engineering imperatives and respond to evolving marketplace situations and customer needs. Writing a perfect first requirement is insufficient if its evolution isn't well managed—poorly controlled change can lead to inadequate systems and software, rework effort and loss of revenue.

Organizations need to implement a reliable and repeatable change control process that helps turn this challenge into an opportunity. As a result, they'll be more competitive, control schedules and respond to evolving customer needs.

Step 8: Capture and track metrics and trends.

Today's complex projects demand automated data collection and reporting facilities to streamline project management. As such, project managers and all stakeholders need a "management dashboard" of metrics and trends that enables them to quickly monitor project activities such as the progress, growth and volatility of actual requirements. In other words, project managers need to keep their focus on decision making instead of manually gathering data and compiling reports. Most importantly, the display of key requirements monitoring information must be at a high level, allowing users to manage by exception and spot trouble areas quickly. A high change frequency on a specific requirement or a whole subsystem may indicate that the requirement needs to be revisited with the customer. A large amount of rework on implementation may point at a poorly specified original requirement.

Trends should also be used to learn lessons from past systems and software projects: Could issues and problems have been identified earlier on? This wealth of information must be used to build the organization's knowledge database (see below). Tracking and analyzing trends is a key practice of CMMI levels 4 and 5, leading to continually improving corporate guidelines for writing better requirements.

Step 9: Provide examples of good requirements.

By providing examples and counterexamples of good requirements and documents, organizations can enhance the quality, consistency and completeness of their requirements. These can originally be templates, industry standards and rules inside a repository, or a corporate intranet.

The next step should be to use good (and bad) requirements from each project that reflect the organization's domain expertise to build a corporate knowledge database. Textbook requirement examples rarely reflect a company's needs as well as their own previous experience. Past requirements should be annotated during a project postmortem to indicate any notable information (positive or negative). New projects can, for example, examine the traceability that previous projects have used for regulations to understand how they were taken into account, and to identify teams that have already achieved compliance for their projects.

Step 10: Reuse requirements.

When a good requirement has been written for a previous project and it is applicable to a present situation, the natural reaction is to reuse it, generally by copying and pasting the description. This unfortunately breaks the traceability and eliminates impact analysis. A smarter approach to reuse is to maintain a link between the two requirements (for example, creating a reuse type link). This enables analysts to access the original requirement at any time to check allocation of implementation, for instance. Likewise, any changes made to the original requirement (issues detected, updates needed) can lead to the notification of reusing teams.

By implementing smart requirements reuse, organizations can improve knowledge sharing across teams and facilitate impact analysis.

Conclusion

Requirements definition and management are among the most important activities in any project, and efforts in this direction can improve and accelerate ROI. It is also the first process improvement area to focus on, based on the “garbage in, garbage out” rule: If the requirements are not clear, any other effort may just help you produce the wrong product faster.

The first step to better requirements management is to understand the simple rules that make a requirement “good.” Training courses and guidance can help organizations achieve this goal.

Once the basic rules are in place, organizations can further increase the quality of their requirements by implementing today’s best practices. These process improvement steps are greatly aided by implementing a requirements management product that not only helps projects to manage requirements more effectively, but also helps future projects benefit from past and current lessons.

For more information

To learn more, please visit:

www.telelogic.com



© Copyright IBM Corporation 2008

IBM Corporation
Software Group
Route 100
Somers, NY 10589
U.S.A.

Produced in the United States of America
October 2008
All Rights Reserved

IBM, the IBM logo, ibm.com, and Rational are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trade-marked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at ibm.com/legal/copytrade.shtml

Microsoft is a trademark of Microsoft Corporation in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

References in this publication to IBM products and services do not imply that IBM intends to make them available in all countries in which IBM operates.

The information contained in this documentation is provided for informational purposes only. While efforts were made to verify the completeness and accuracy of the information contained in this documentation, it is provided "as is" without warranty of any kind, express or implied. In addition, this information is based on IBM's current product plans and strategy, which are subject to change by IBM without notice. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this documentation or any other documentation. Nothing contained in this documentation is intended to, nor shall have the effect of, creating any warranties or representations from IBM (or its suppliers or licensors), or altering the terms and conditions of the applicable license agreement governing the use of IBM software.

IBM customers are responsible for ensuring their own compliance with legal requirements. It is the customer's sole responsibility to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer's business and any actions the customer may need to take to comply with such laws.