Requirements and quality management solutions
White paper
June 2009

Rational. software

An integrated approach to requirements and quality management.

The principles of requirements-driven testing

Contents

Executive summary

Requirements management plays a critical role at virtually every stage of project development. Development teams that can consistently design projects using precise, measurable requirements, and gauge their projects' progress toward meeting those requirements, can vastly improve their efficiency, agility and marketability.

In this white paper, we'll outline the many benefits of an integrated approach to requirements and quality management, as well as offer best practices for effective requirements-driven testing.

We will also review the key principles to keep in mind when integrating requirements-driven testing into your project's lifecycle, and discuss the merits and evolution of classic testing models.

What is requirements management?

A requirement is an assertion that must be satisfied by a product or process.

There are many kinds of requirements and many words that may be used to define the term, including: aim, aspiration, condition, contract, constraint, goal, mandate, need, objective, obligation, regulation, requisite, rule, specification and target. In this paper, we will use the word requirement as a synonym for all of these concepts. We will also use it to refer to individual statements, clauses and items rather than entire requirements documents.

To effectively manage requirements, companies must be able to design clear, testable requirements, understand their importance within a project and trace them throughout a project's lifecycle.

Requirements management is the set of disciplines and activities concerned with the capture, formulation, organization, versioning, publishing, tracing, analysis, and change of requirements.

Many people mistakenly believe that requirements management is something that takes place during the definition stages of a project and is then complete. In fact, requirements exist in some form at virtually every stage of development. Indeed, requirements play a vital in the many stages of testing, even those at the end of the development process.

Highlights

In this paper, we will focus on three key requirements management disciplines:

- Requirements expression. *Not only should requirements be concise and unambiguous, they should also be testable. This usually means making sure that requirements are properly quantified and that test criteria can be clearly identified. This test criteria provide essential information used during testing.*
- Requirements classification. *Not all requirements are created equal; some requirements have far greater significance than others. To manage project development effectively, each requirement statement must be classified appropriately for the application to aid in effective decision making and resource allocation*
- Requirements traceability. *Ensuring traceability is a vital part of requirements management. Developers and managers must be able to track requirements through the stages of development. Likewise, they must also be able to trace tests back to their associated requirements to effectively track progress and analyze the impact of changes.*

Testing throughout the development cycle helps pinpoint defects and trace which requirements have and haven't been met.

What is testing?

Testing is any activity aimed at revealing defects.

This is a deliberately broad definition of testing. It covers many early activities such as requirements reviews, design inspections, structured walkthroughs and analysis from modeling as well as those in the traditional unit, system, and acceptance testing stages.

For many people, this definition may be too broad because it removes testing from its traditional place on the right-hand side of the V-model. But this definition includes the concept that requirements and designs themselves can be "tested" (reviewed, validated, verified, qualified or evaluated) before anything is built or coded.

In keeping with the breadth of this definition, we use the word *qualification* to cover testing, validation, verification and evaluation. Thus "test criteria" become "qualification criteria," and "test plans" become "qualification plans."

Highlights

A defect is any departure from a requirement.
Defects are recognized, ideally, with reference to a requirement. There are requirements directly for the product (originating from customers) and requirements for the development process (standards and procedures). The latter includes best practices, which are themselves designed to reduce the likelihood of introducing and propagating defects.

Requirements-driven best practices call for creating the requirement and its test at the same time.

Principles of requirements-driven testing

There are several core principles of testing that should always be respected. They represent, at a high level, elements of best practice in requirements-driven testing:

Plan tests early

Plan tests for each requirement as the requirement is written.
Considering how to test a requirement when you are capturing it can improve the way you express the requirement because the process encourages you to consider how the requirement can be quantified. For each requirement ask the question, *"How will we know if this requirement will be, or has been, satisfied?"*

Note the use of both the future and past tense in this question. Early tests such as design inspections verify that, when the product is built according to the design, it will meet requirements. Later tests check that requirements *have been* satisfied by what was built.

The above question should lead to a *qualification* strategy for each requirement. Because of the broad range of qualification activities available, nearly every requirement will spawn multiple tests covering all stages of development. For instance, an early opportunity for qualification is to check the design against a requirement. Are the necessary elements present in the design to ensure that the requirement will be met? In addition, there will likely be one or more system tests to ensure that what is finally built actually meets the requirement. The set of selected tests form the qualification strategy. .

Highlights

The question should also lead to the identification of *qualification criteria* for every test. These criteria formalize the definition of a successful outcome for each test. Qualification criteria vary with the nature of the requirement and the planned test.

Testing for defects at the earliest possible point in a project's lifecycle helps minimize costly rework.

Conduct tests early

Perform tests as early as possible in the development process.
An old saying advises, "a stitch in time saves nine." Defects propagate from stage to stage, and the cost of correcting them geometrically increases throughout the development process. You can save both time and money by identifying and correcting defects as early as possible, thus minimizing expensive rework in later stages.

It's important to have at least one test for each requirement and the ability to understand what impact changing a requirement will have on the entire system.

The qualification plan for each requirement should therefore define the earliest possible means of detecting defects. Such means might include reviews, inspections and walkthroughs.

Relate tests to requirements

Trace tests back to the requirements they are designed to check.
Tracing is a means of documenting the relationships between artifacts during development. The most common form of tracing tracks the *satisfaction* relationship between layers of requirements. However, it's also important to trace the qualification relationship between requirements and the tests that show that they have been met.

Tracing enables two kinds of analysis:

Effective testing tests the individual components and their interactions with the rest of the system.

- Coverage analysis *helps you ensure, for example, that every requirement has at least one qualification action planned for it and that the action is ultimately carried out. It can also help ensure that every qualification action has an associated requirement and, therefore, provides benefit.*
- Impact analysis *can help you determine which qualification actions need to be taken when requirements change. Before accepting a change to a requirement, it's important to assess the cost of redefining and re-executing its associated qualification actions. Similarly, it's important to understand how a failed test may impact the requirements qualification actions. Similarly, it's important to understand how a failed test may impact the requirements.*

Highlights

Each defect is tied to one or more requirement. A good requirement management system enables you to track your project's progress toward meeting the defined requirements.

Impact analysis uses a combination of the *satisfaction* and *qualification* relationships to determine potential rework. For instance, if a component test fails, it not only affects the associated component requirements (through the *qualification* relationship) but also the sub-system requirements that those component requirements are supposed to satisfy (through the *satisfaction* relationship) and, in turn, the satisfied system and customer requirements.

Testers understand that testing at every level is necessary. For instance, a sub-system requirement cannot be fully tested by simply testing its components; the sub-system (the integration of those components) must also be tested. This aligns with the concept of *emergent properties* in requirements management: a system is more than the sum of its parts, and some behavior arises from the interaction of components.

Relate defects to requirements

Trace defects back to the requirements that they show have not been satisfied.
When a test reveals unexpected behavior, or that test criteria are not met, defects arise. There are three possible causes of defects:

1. *An error in the definition of the test.*
2. *An error in the expression of the requirement or in its qualification criteria.*
3. *A genuine defect in the product.*

Defects are identified by tracing them to one or more unmet requirement. In fact, the definition of "defect" is "a departure from requirements." Since each test may be traced to multiple requirements, you must analyze each defect to determine precisely which requirements are affected.

Highlights

Measure progress against requirements

Set targets and measure your testing progress by which requirements are satisfied or not satisfied.

One of the most difficult judgments to make during development is deciding when to stop testing. With limited resources available, the test manager needs to know where additional testing efforts will be most effective. If testing is isolated from requirements, the manager cannot judge the relative importance of different aspects of the system and cannot make effective decisions about resource allocation or the impact of failures

With the appropriate traceability in place, it's easy to understand the relative importance of each test and how the success or failure of tests can affect requirements at each level—customer requirements, system requirements, etc. Traceability also enables project leaders to set reasonable targets and gauge progress, allocate resources where they are most needed and generate accurate reports that illustrate the test results in relation to requirements.

The W-model expands the classic V-model to illustrate relationships between requirements and actions.

The W-model

A process and data model called the W-model, an evolution of the classic V-model illustrates how we might best implement these principles of requirements-driven testing.

In the classic V-model, shown in Figure 1, the rounded boxes represent activities through a project's lifecycle. (This model should not be interpreted as a "Waterfall" model, since all these activities may take place in parallel, with constant feedback.)

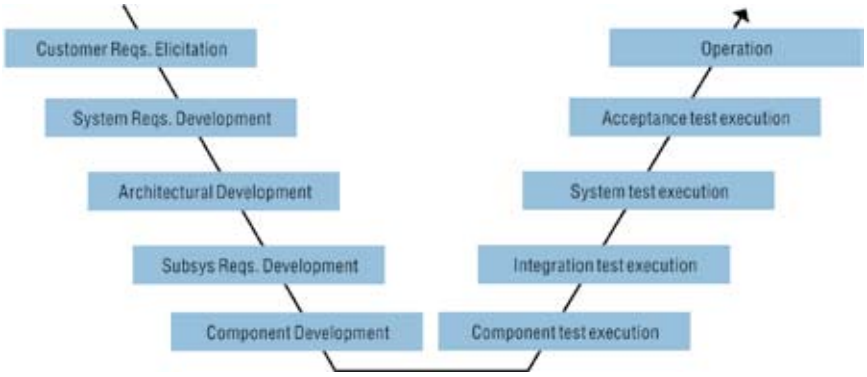

Figure 1: The classic V-model

Highlights

Qualification schedules depict a timeline for each unit test, review or other qualification action at each development stage

Figure 2 adds the data produced by the activities, represented by rectangles, and the tracing between the data, represented by the thick arrows. This model places qualification plans parallel to the requirements on the left-hand axis of the V-model. The qualification plans contain only the planned qualification actions and the qualification criteria. The result of each action is collected when the action is executed at the appropriate stage of development.

You can use tracing to document two relationships:

1. *Satisfaction* between layers of requirements.
2. *Qualification* between the planned qualification actions and requirements.

Schedule each qualification action (review, inspection, unit test, integration test, etc.) identified in the qualification plans to occur at the appropriate stage of development. Collectively they form a "Qualification Schedule" for the whole project.

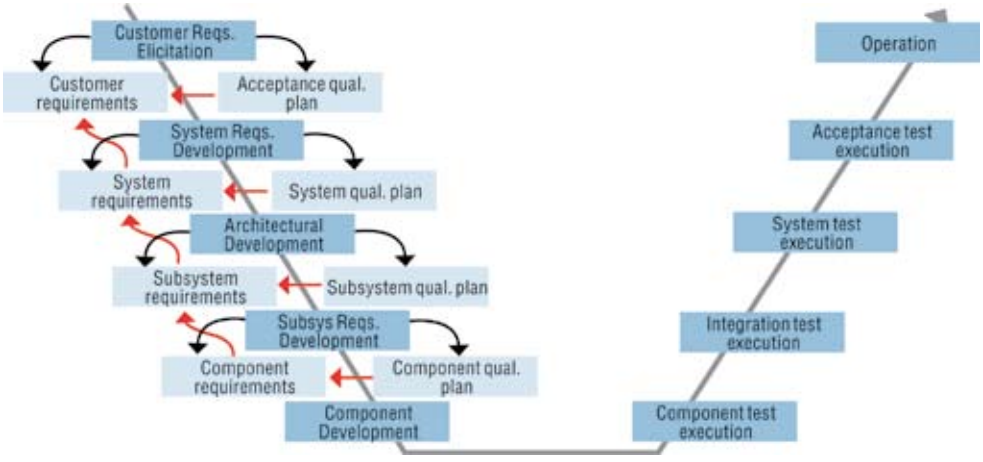

Figure 2: The V-model with qualification plans

Highlights

Defect information may be stored in a single repository for further analysis.

When initially designing tests, you often don't have enough information to plan the exact details. Therefore, Figure 3 shows test design as separate activities at each level. These activities lie on a parallel axis to the development axis. Due to its underlying shape, this model is sometimes referred to as the W-model.

To design effective tests, use the qualification plans to devise specific tests that may be traced back to the plans. You may need to design several tests for each plan, or a single test may cover several planned tests.

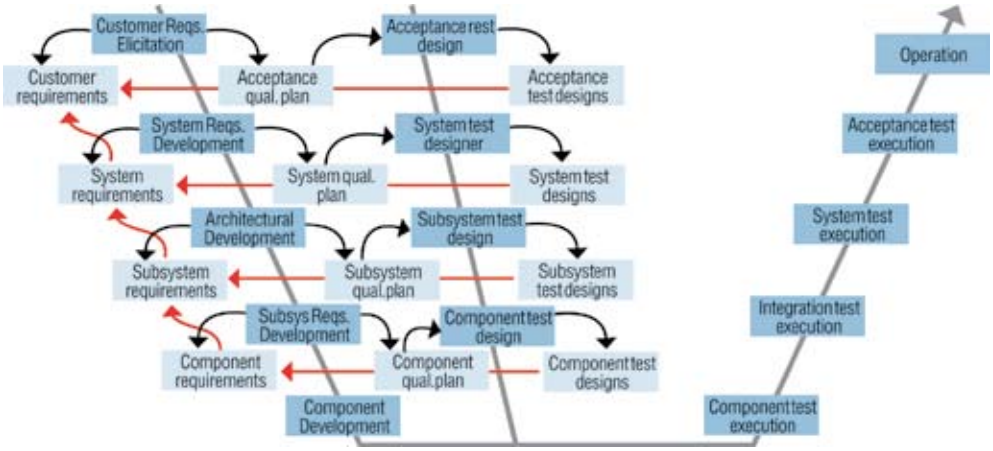

Figure 3: The V-model (or W-model) with test designs

The W-model shows test design as separate activities on each level.

Figure 4 shows the model with the test result data. In this figure you can see status and defect information.

Defects are placed in a single repository, since there is no specific assignment of defects to layers of development. Test execution activities include analyzing defects to determine whether there is a genuine product defect and which requirements are affected. Figure 5 illustrates the result of this analysis.

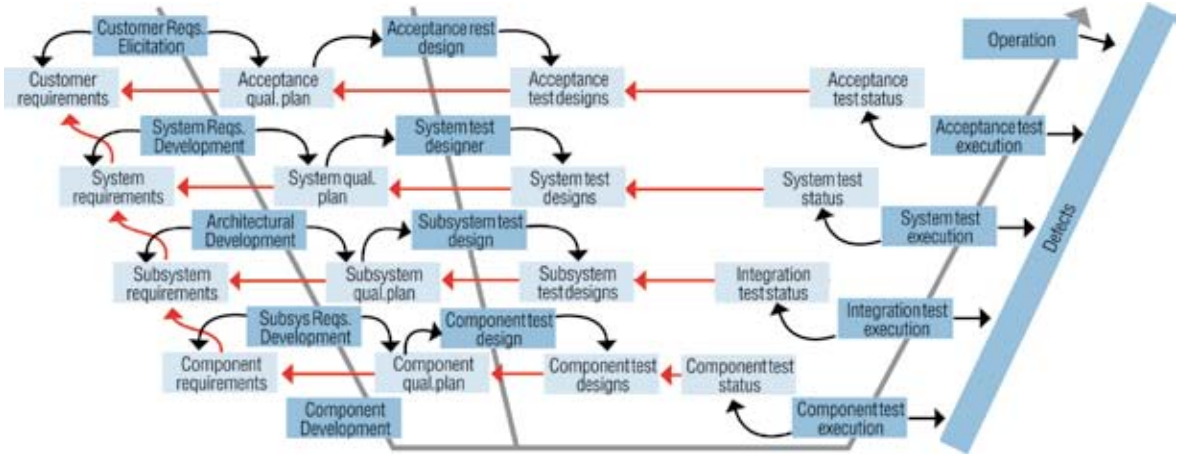

Figure 4: The W-model with test results

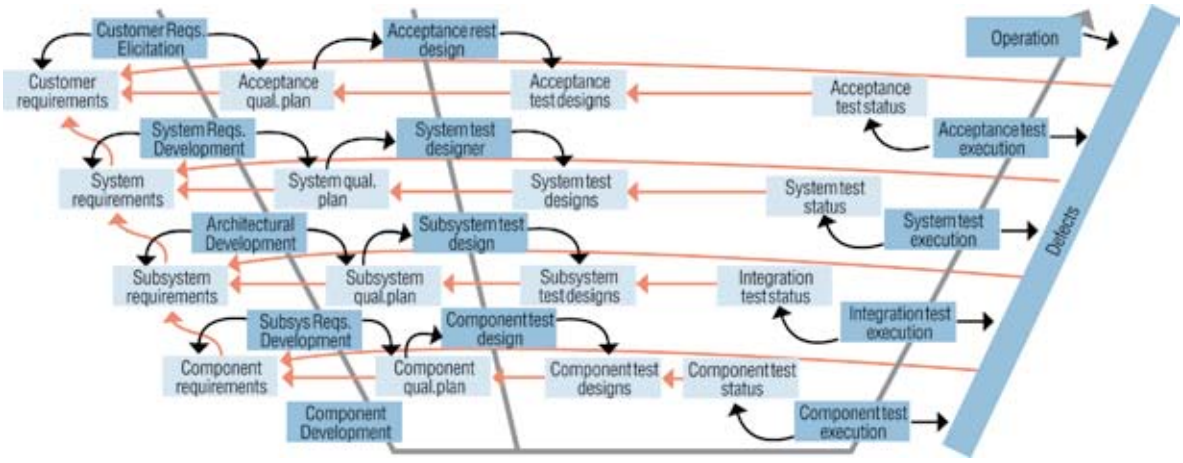

Figure 5: The W-model with defects traced to requirements

Highlights

Analyze the defects to determine which requirements are affected and what action(s) must be taken.

This approach accommodates the three principles outlined above as follows:

- Plan tests early. *Writing qualification plans in parallel with the requirements, ensures that consideration is given to testing as the requirement is written.*
- Conduct tests early. *Mapping qualification plans onto the Qualification Schedule encourages creating a timeline for qualifications and determining the earliest point to perform tests on each requirement.*
- Trace test and identified defects back to the requirements. *The tracing model allows this relationship to be maintained and analyzed.*

Note that, to determine whether an individual requirement has been fully realized, both the satisfaction and qualification relationships are used to gather a complete set of test results.

Iterative testing helps ensure all problems have been identified, classified and resolved, as needed.

Finally, testing tends to be highly iterative; you perform tests, you find defects and correct them and you repeat the tests. Figure 6 represents multiple test runs against the same criteria. Through the sequence of test runs, the test results evolve and progress may be determined.

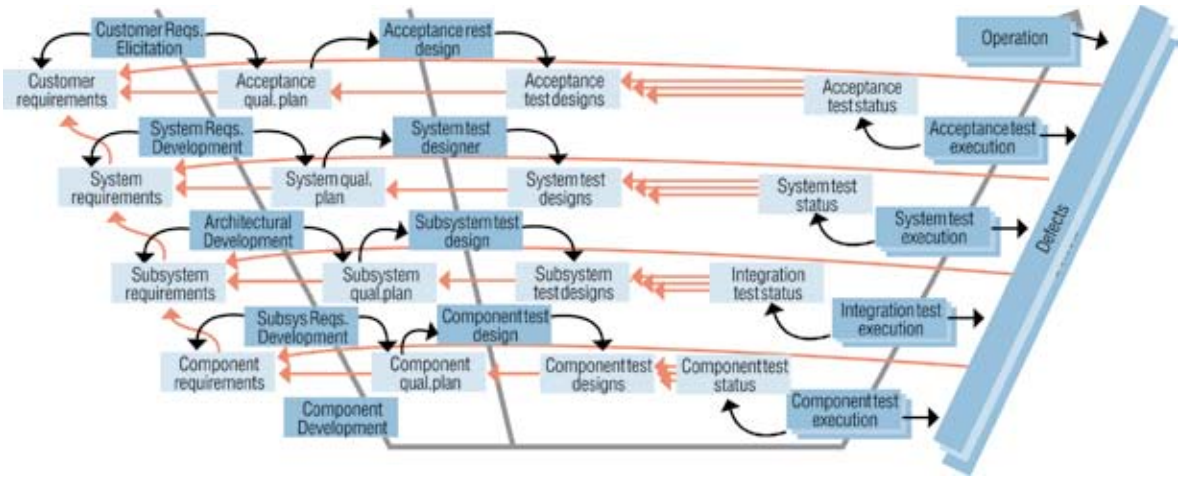

Figure 6: The W-model with multiple test runs

Conclusion

To achieve a sustainable competitive advantage in the marketplace, today's product, system and software development organizations must look beyond individual productivity and single- point solutions.

Implementing requirements-driven testing using integrated tools can help organizations synchronize test and development data among analysts, developers and testers. Sharing a common understanding of test results and the relationship between requirements can enable organizations to boost productivity and product quality and better meet the demands of their customers both today and in the future.

IBM® Rational® DOORS® and IBM Rational Change offer the integrated toolset companies need to create tests, trace requirements and manage defects throughout a project's lifecycle.

For more information

To learn about IBM solutions for integrated requirements management and quality management, please contact your IBM sales representative or IBM Business Partner, or visit: **ibm.com**/software/rational

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

Produced in the United States of America
June 2009

RAW14142 -USEN-00