

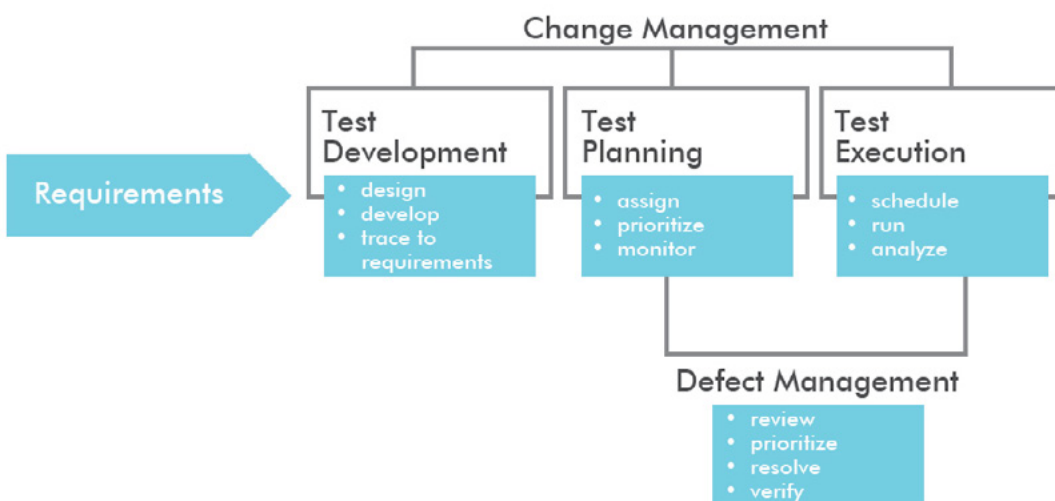
Requirements-Based Testing: Encourage Collaboration Through Traceability

Executive Summary

It is a well-documented fact that incomplete, poorly written or poorly communicated requirements are responsible for 50 to 70 percent of software project failures. Indeed, industry research shows that the root cause of 56 percent of all errors identified in projects are introduced in the requirements phase.¹

As a result, many organizations are looking to reduce their risk by implementing formal requirements management practices. The adoption of a formalized requirements management discipline can be leveraged by the quality assurance organization. By implementing a requirements-based testing methodology and ensuring complete requirements test coverage, companies can increase efficiency, reduce the risk of failure and improve overall software quality. Requirements-based testing is also popular with organizations following iterative or agile development styles, because it ensures complete coverage of changing requirements, in which a continuous develop-build-deploy cycle is in use.

¹ "What Is Requirements-Based Testing?" – Gary E. Mogyorodi, Bloodworth Integrated Technology, Inc., STSC Crosstalk, March 2003



In this white paper, you will learn how Integrity, a PTC product that is a unified platform for application lifecycle management, can help your QA team implement a requirements-based testing process that enables staff to:

- Know what to test and ensure all project requirements have complete test planning and test execution coverage
- Establish traceability between all artifacts in the application lifecycle, such as requirements, specifications, test cases and code
- Manage the resolution of defects identified during testing
- Measure project readiness through reports and dashboards incorporating real-time metrics and trends

Description of Requirements-Based Testing

A requirements-based testing methodology begins with requirements management. Business analysts, business users, product managers and other business professionals compile and manage requirements and specification documents as an accurate and complete description of the needs of the system. The QA department relies on these requirements as a basis to plan and execute its testing efforts. By tracking test planning and execution of test cases directly against the requirements,

QA can confidently measure progress as the project evolves and ensure all requirements have been validated. When all requirements have been thoroughly tested, and the defects identified during testing have been resolved, the QA department can provide formal release sign-off.

A requirements-based testing methodology helps identify problems early in the application lifecycle, which can substantially contribute to development cost savings. The cost of fixing a software error rises dramatically as a software project progresses. If an issue is caught during the requirements phase, the cost ratio to fix the error is one to one; at coding, it is seven to one; and at production, it ranges from 40 to one to 100 to one.²

² B. Boehm and V. Basili, "Software Defect Reduction Top 10 List," – IEEE Computer

Cost to find and fix a requirements defect	Phase	Average Cost of Repair Defect	Normalized Value
	Requirements	\$139	1
	Design	\$455	3
	Coding	\$1,000	7
	System Testing	\$7,000	50
	Production	\$14,000	100

The Challenges with Current Testing Methods

Manual Processes

Many challenges exist when using manual methods to manage testing within the QA department, including:

- Inconsistent quality and problems passing audits, due to lack of enforcement for processes or standards across multiple projects
- Time wasted manually preparing reports and evaluating the status of testing
- Inability to ensure validation of requirements because no direct link to test cases exists
- Project disruption caused by changes to requirements when inadequate impact analysis is done
- Errors and omissions caused by incomplete test coverage against requirements
- Inaccurate testing as a result of requirement churn during the project lifecycle

Multiple Point Products and Silos

Over time, many organizations incrementally adopt point products to solve specific team-based needs. Implementing multiple products for requirements management, test management and defect tracking creates disconnected silos of information, resulting in the following collaboration challenges:

- There is no single source of reporting available to management that spans all phases of the application lifecycle
- It is difficult to implement business rules and enforce processes across domains
- Changes that take place in one domain are not accounted for in another

Requirements-Based Testing with Integrity

A unified platform for application lifecycle management, such as Integrity, can help QA organizations conform into lockstep with all other phases of the application lifecycle and also help them overcome the challenges presented by manual or multiple point system approaches.

Change Management

Managing change effectively throughout the application lifecycle is critical for success. Process controls within the Integrity platform ensure that only authorized individuals make changes at the appropriate time. Comprehensive impact analysis can be performed to assess the risk of accepting a change. When changes are accepted, related items that may be impacted are flagged for review, ensuring appropriate corrective measures are taken.

Connecting the Requirements Phase to Downstream Testing Efforts

Most organizations today define their requirements in business tools such as Microsoft Word, Excel or Project. These requirements documents can be easily imported into Integrity. Alternatively, users can take advantage of Integrity's document-centric interface to author requirements directly within the system.

Workflow within Integrity then initiates an approval cycle for these requirements and, upon their approval, test authors can create test cases linked directly to the requirement, just as developers do for functional specifications. Through its direct, downstream connection with development and QA, Integrity enables proposed requirements changes to be analyzed for impact and ensures that all changes made to requirements are communicated downstream.

A detailed overview of Integrity's requirements management capabilities is provided in the white paper, "An Innovative Approach to Managing Software Requirements."

Plan Testing with Requirements in Mind

Now that the software application is well defined by a set of requirements, the QA organization can implement a focused testing effort that is based directly on the requirements, to ensure that the system is being verified against original intent and that it will deliver on the business need. This next step begins with test development. Test authors prepare one or more test cases for each requirement to ensure the requirement will be met in the final deliverable. Test developers enhance the test case to include details regarding the test steps a test engineer will perform, as well as the expected results and pass/fail criteria that will be used to evaluate the test. For automated testing, a link is established to the test script in the external automated test execution product.

By linking requirements to test cases, a feedback loop between business analysts and QA is established. This feedback loop helps refine the requirements and, in general, improves the product. By asking questions to clarify precisely what needs to be tested, errors or gaps in the requirements may be uncovered.

Conduct Automated and Manual Tests with Confidence

Once test cases are prepared and logically grouped together, a QA manager will assign testing activities to test engineers, indicating that, they are ready to run. In the case of automated testing, Integrity interacts with external test execution products such as HP Quick Test Professional and ITKO Lisa through its Automated Test Execution Framework.

Integrity connects with the external system, provides it with information needed to run the tests, and then captures the results. The test results are published in Integrity for review and interpretation. When running manual tests, test engineers use the Test Results Editor. They are provided with a focused view of the test steps needed to be run, the expected results and the pass/fail criteria. Detailed comments on the results can be logged. The test engineer will also review the results of automated testing in this Editor.

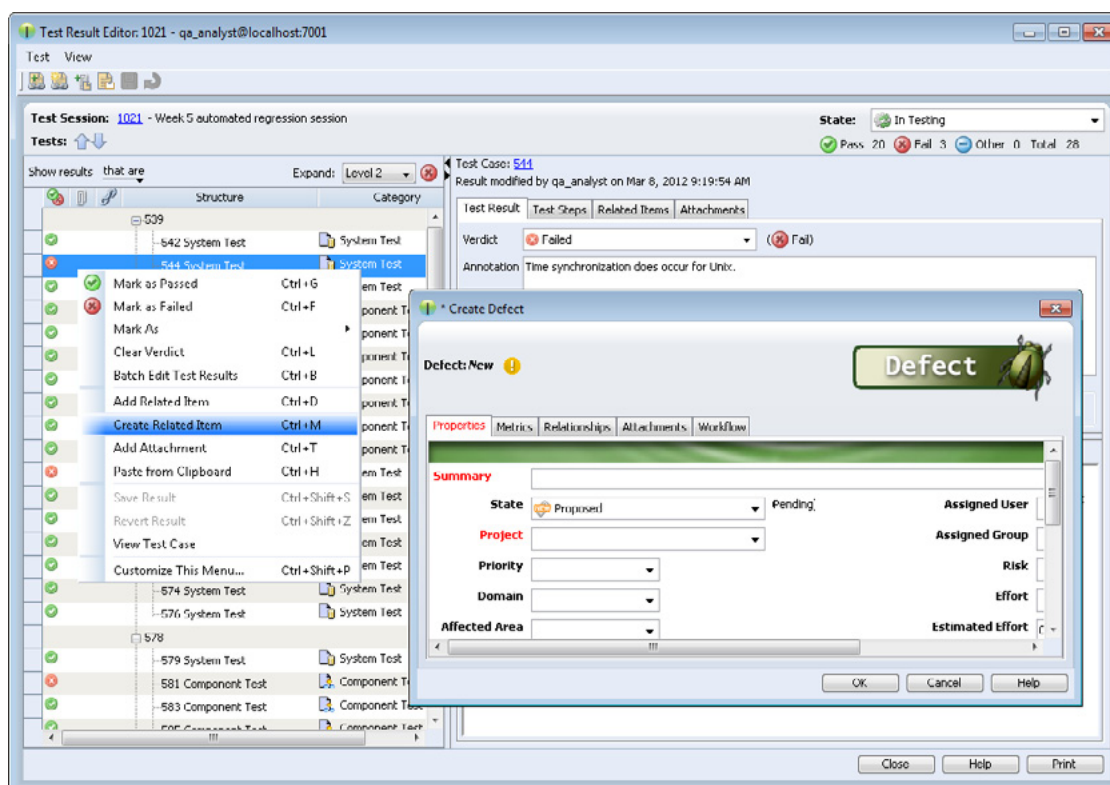


Figure 1: Create a defect linked to a failed test case and its corresponding requirement.

Resolve Defects and Lower Project Costs

One of the key measurements of product readiness is the number and severity of open defects. As referenced earlier, the earlier defects are detected and corrected in the development lifecycle, the lower the cost. Defects are prioritized and assigned to a developer for corrective action. The system establishes traceability between the defect and all changes the developer makes to resolve them. This closed loop provides complete traceability from the requirements down to the lines of code changed to repair defects identified during testing.

Charts/Reports/Dashboards

QA managers, project managers and executives need accurate and up-to-date information to plan and schedule resources, take quick corrective action and generally manage schedules and budgets with confidence. Too often reporting is done manually, resulting in time delays. There's also the risk of subjectivity, given the human "collection" factor. Point tools each have their own data repository and therefore cannot provide an aggregate view of the application lifecycle.

Integrity includes extensive graphing, charting and reporting capabilities. Role-specific dashboards display the real-time information team members need to do their jobs. Information can be displayed for a single project or filtered to include all the projects of interest. A single click provides drill down into the lifecycle details behind the chart or report. For example, if a QA manager sees three high-priority defects outstanding, she will be taken directly to those defects by clicking on that section of the chart, thereby allowing her to further investigate and take quick corrective action.

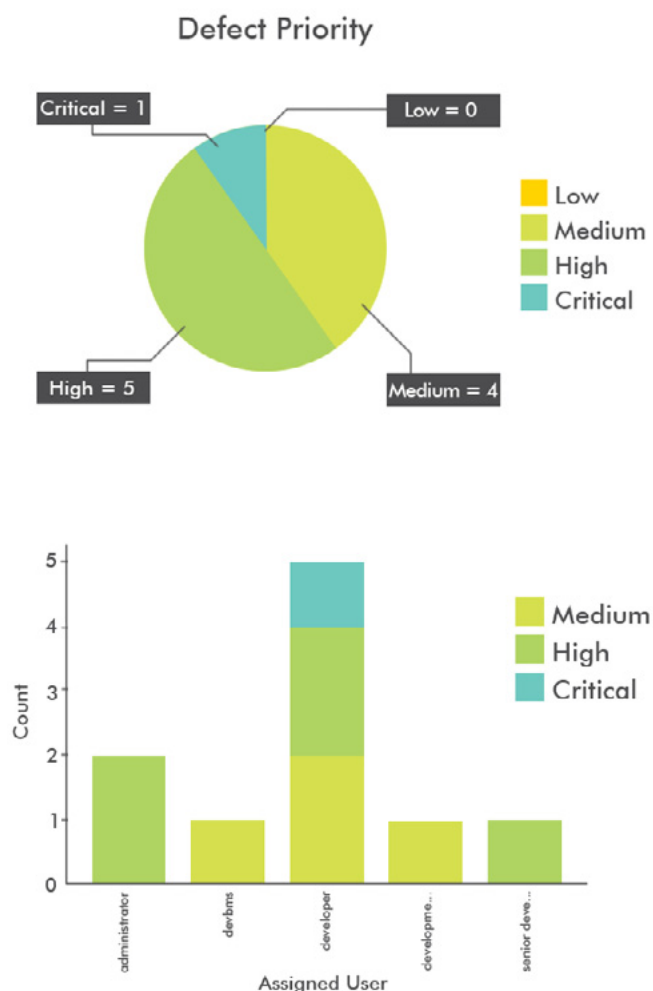


Figure 2 & 3: Charts provide a real-time view into critical testing metrics.

Summary

By implementing automated requirements-based testing processes within Integrity, your QA organization can help ensure that software projects approved for release are on time, and on budget and meet the needs of the business by:

- Assuring requirements receive complete test planning and test execution coverage
- Reducing disruptions that arise due to requirement churn
- Accessing a single source of real-time information to measure project readiness
- Ensuring processes and standards are enforced across all projects
- Establishing traceability between all artifacts and visibility across the entire development lifecycle

Integrity Business Unit Locations

North America
1 800 613 7535

United Kingdom
+44 (0) 1252 453 400

Germany
+49 (0) 711 3517 750

Asia Pacific
+65 6830 8338

Japan
+81 3 5422 9503

integrityinfo@ptc.com

For more information visit: [PTC.com/products/integrity](https://www.ptc.com/products/integrity)

© 2012, Parametric Technology Corporation (PTC). All rights reserved. Information described herein is furnished for informational use only and is subject to change without notice. The only warranties for PTC products and services are set forth in the express warranty statements accompanying such products and services and nothing herein should be construed as constituting an additional warranty. References to customer successes are based upon a single user experience and such customer's testimonial. Analyst or other forward-looking statements about PTC products and services or the markets in which PTC participates are those of the analysts themselves and PTC makes no representations as to the basis or accuracy thereof. PTC, the PTC logo, and all PTC product names and logos are trademarks or registered trademarks of PTC and/or its subsidiaries in the United States and in other countries. All other product or company names are property of their respective owners. The timing of any product release, including any features or functionality, is subject to change at PTC's discretion.

7513-Integrity: Requirements-Based Testing-WP-EN-0312