TechExcel

5 Steps to Improve Product Quality Through Requirements Traceability

Executive Summary

Quality Management is more than just testing & bug fixing. To deliver a quality product, especially when both hardware and software are involved, the product must not only be bug-free, but must also provide the features, functionality, and user experience your customers require. To further complicate matters, these requirements often evolve and change from the beginning of a release cycle to the end. Therefore, a comprehensive Quality Management process must include both testing for bugs, but also verification testing that all current requirements intended for a release are in fact present and working properly.

This white paper will explore the challenges and five key best practices to consider when implementing a comprehensive Requirements Traceability Process and how to leverage this approach to ensure success.

Whether your development process is Agile or more traditional, one of the biggest challenges in product development is ensuring that what you actually deliver matches your most current requirements. Whether developing simple manufactured items, software products or systems, or more complex products with hardware, firmware, and software components, ensuring that your final (and potentially evolving) design matches the finished product requires a well-defined process. When software development is linked to hardware & firmware development, it is especially challenging because of the wide range of requirements and multiple dependencies that are present for relatively simple applications. It is for this reason that a well-designed Requirements Traceability process is critical to delivering a quality product.

1 Linking Requirements to Development and Testing

To ensure that the released product delivers what was intended, it is essential to link the underlying requirements to the planning, development, and testing processes. It is also important to realize that these requirements can take many forms, including:

- Customer Requirements
- Internal Requirements
- Marketing Requirements
- Performance Requirements
- Compatibility Requirements
- Regulatory Requirements

While some requirements may be optional, others, such as compatibility, regulatory, and customer requests may be critical to the product's success. Differentiating between these categories is key when designing a comprehensive Requirements Traceability process.

Once you have defined the types of requirements appropriate to your development project, it is then important to understand the various development artifacts you will be tracking.

Not every project will need to track every type of artifact, but depending on which methodology you choose (Agile, Waterfall, hybrid approaches) these artifacts might include:

- Design Specifications
- Product Backlog Items
- Release / Sprint Backlog Items
- Interim Releases / Builds
- Development Tasks
- Source code
- Code Reviews
- Test Cases
- Bugs
- Product Documentation

The challenge then is to keep these artifacts up to date as the underlying requirements evolve and change from the start of the project to the final release.

2 Defining Requirements Traceability

Requirements Traceability refers to the mapping of product requirements to the key artifacts created throughout the development process. Once this mapping is done, and the requirements and artifacts are linked, it becomes easy to validate the design, implementation, testing, and bug fixing for each individual requirement. This will ultimately ensure comprehensive product quality and market success.

For small development teams, links between requirements and test cases might be all that is needed, with design and development managed independently. But for larger projects, especially those with hardware and software components, it is critical to connect the requirements to each phase of the development process. This not only ensures that each requirement will be implemented correctly, but it also allows changes to the underlying requirements to flag linked design docs, implementation tasks, test cases templates, testing tasks, pending bug fixes, etc. Each artifact can then be reviewed and updated if necessary based on the revised requirement. Without such traceability links it becomes difficult to impossible to coordinate evolving requirements with the ongoing development project.

The diagram below illustrates the inherent complexity of linking a wide variety of requirement types with the various artifacts created through design, development, and testing. A well designed Requirements Traceability process helps to make these complex relationships manageable.

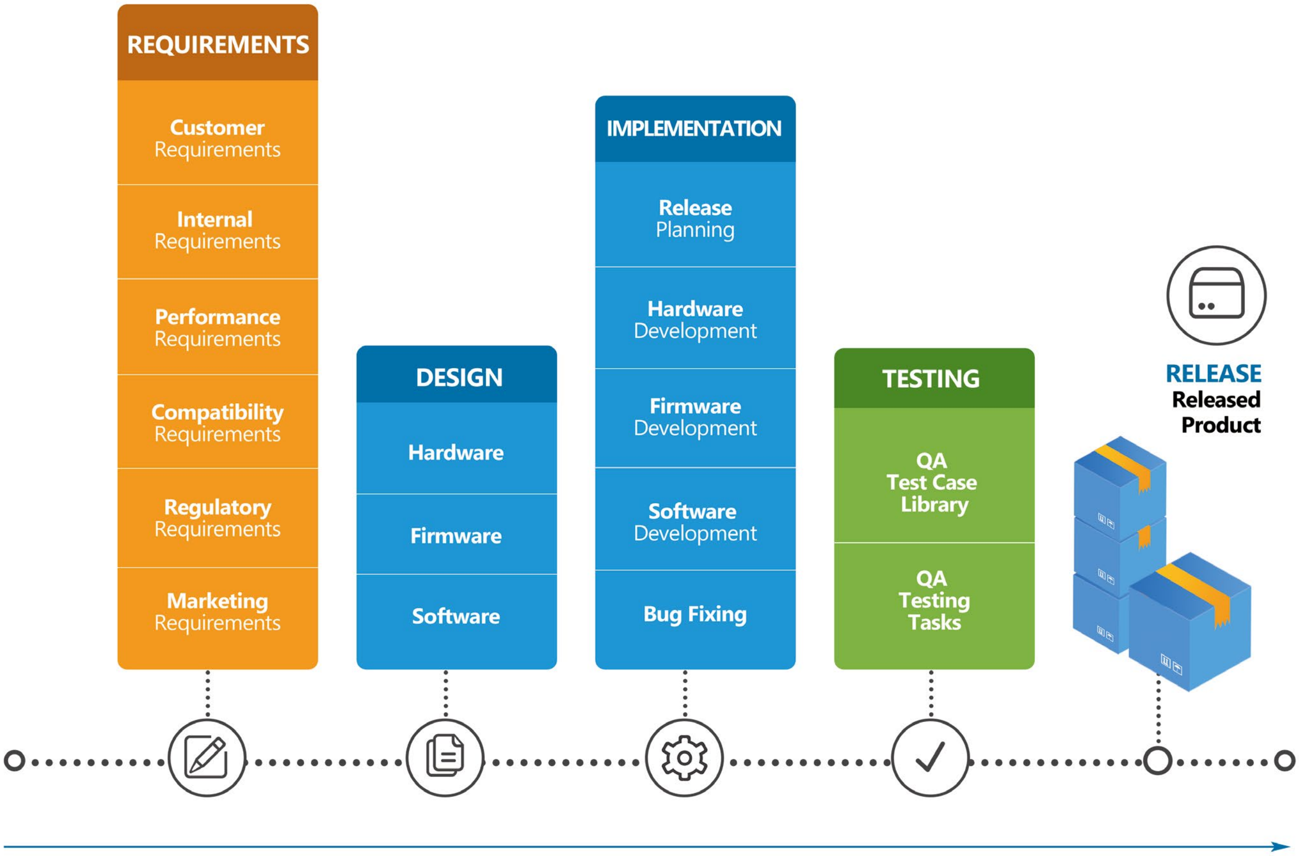


Figure 1 Requirements and Developments Artifacts

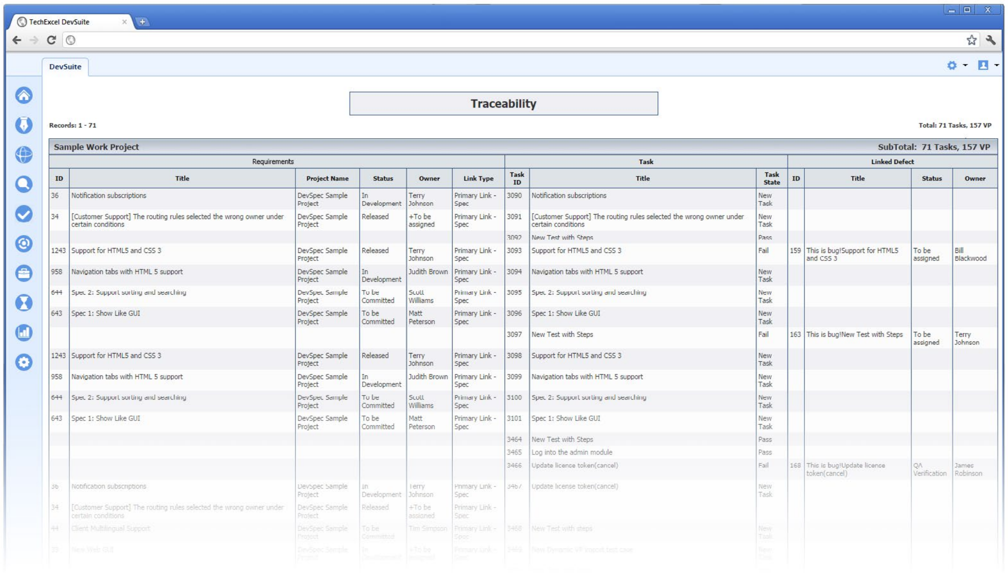
3 Implementing Requirements Traceability

When planning the implementation of a Requirements Traceability process, it is important to first evaluate your various development artifacts and decide which of them should be linked to the supporting requirements. If requirements and artifacts are managed in several different systems, the process of linking them can be quite time consuming and difficult. That being said, when multiple systems are being used within a development project, it makes a well-defined traceability process even more important. With multiple teams involved, it increases the chances that the artifacts will not be properly synced with requirements, and each step in the process creates another chance that details may be lost.

Once you have identified the traceability artifacts critical to your product development process, the next step is to link them to the underlying requirements. When multiple systems are being used for requirements, development, and testing, this may require the use of APIs or custom integration. However when using an integrated Product Development Lifecycle platform, linking requirements to artifacts is often a built-in capability. In either case it is important that the traceability links are dynamic in nature, and will flag each artifact for review when the linked requirement is updated. It is rare that the requirements for any product development project remain unchanged from kickoff to release, and only with dynamic traceability links can you ensure that these changes are fully reflected in the final released product.

4 Traceability Reporting

Dynamic traceability links allow your teams to keep up to date with evolving requirements, but they also provide key data for one of the most important aspects of any product development project, the Traceability Report. Essentially such a report will confirm that for each requirement the appropriate development, testing, bug fixing, re-testing, and final approvals have all been completed and the product is ready for release. It is important to reiterate that releasing a quality product is more than just verifying that it is bug-free. Often it is more important to validate that the complex set of requirements you intended to include in the product are in fact present in the final release. A comprehensive traceability report is the best way to achieve this validation.



Traceability

Records: 1 - 71

Total: 71 Tasks, 157 VP

Sample Work Project

SubTotal: 71 Tasks, 157 VP

Requirements						Task			Linked Defect			
ID	Title	Project Name	Status	Owner	Link Type	Task ID	Title	Task State	ID	Title	Status	Owner
36	Notification subscriptions	DevSpec Sample Project	In Development	Terry Johnson	Primary Link - Spec	3090	Notification subscriptions	New Task				
34	[Customer Support] The routing rules selected the wrong owner under certain conditions	DevSpec Sample Project	Released	+To be assigned	Primary Link - Spec	3091	[Customer Support] The routing rules selected the wrong owner under certain conditions	New Task				
						3092	New Test with Steps	Pass				
1243	Support for HTML5 and CSS 3	DevSpec Sample Project	Released	Terry Johnson	Primary Link - Spec	3093	Support for HTML5 and CSS 3	Fail	159	This is bug!Support for HTML5 and CSS 3	To be assigned	Bill Blackwood
958	Navigation tabs with HTML 5 support	DevSpec Sample Project	In Development	Judith Brown	Primary Link - Spec	3094	Navigation tabs with HTML 5 support	New Task				
644	Spec 2: Support sorting and searching	DevSpec Sample Project	To be Committed	Scott Williams	Primary Link - Spec	3095	Spec 2: Support sorting and searching	New Task				
643	Spec 1: Show Like GUI	DevSpec Sample Project	To be Committed	Matt Peterson	Primary Link - Spec	3096	Spec 1: Show Like GUI	New Task				
						3097	New Test with Steps	Fail	163	This is bug!New Test with Steps	To be assigned	Terry Johnson
1243	Support for HTML5 and CSS 3	DevSpec Sample Project	Released	Terry Johnson	Primary Link - Spec	3098	Support for HTML5 and CSS 3	New Task				
958	Navigation tabs with HTML 5 support	DevSpec Sample Project	In Development	Judith Brown	Primary Link - Spec	3099	Navigation tabs with HTML 5 support	New Task				
644	Spec 2: Support sorting and searching	DevSpec Sample Project	To be Committed	Scott Williams	Primary Link - Spec	3100	Spec 2: Support sorting and searching	New Task				
643	Spec 1: Show Like GUI	DevSpec Sample Project	To be Committed	Matt Peterson	Primary Link - Spec	3101	Spec 1: Show Like GUI	New Task				
						3464	New Test with Steps	Pass				
						3465	Log into the admin module	Pass				
						3466	Update license token(cancel)	Fail	168	This is bug!Update license token(cancel)	QA Verification	James Robinson
36	Notification subscriptions	DevSpec Sample Project	In Development	Terry Johnson	Primary Link - Spec	3467	Update license token(cancel)	New Task				
34	[Customer Support] The routing rules selected the wrong owner under certain conditions	DevSpec Sample Project	Released	+To be assigned	Primary Link - Spec							
44	Client Multilingual Support	DevSpec Sample Project	To be Committed	Tim Simpson	Primary Link - Spec	3468	New Test with steps	New Task				
33	New Web GUI	DevSpec Sample Project	In Development	+To be assigned	Primary Link - Spec	3469	New Dynamic VP import test case	New Task				

Figure 2 Traceability Reporting

The report shown above illustrates how a well-defined requirements traceability process can facilitate effective validation reporting to confirm your product is ready for release.

5 Traceability and Agile Development

Agile development does not typically have a formal requirements management process, but the benefits of requirements traceability are still important. In fact, given the ever-changing product backlog inherent in Agile Development, it is especially important to connect those updates to the testing and release process to ensure you are delivering what you intended. By applying the above traceability process and structure to an Agile Product Backlog and Sprint Backlogs, and linking those backlog items to test cases, similar results can be achieved while preserving the flexibility and dynamic nature of Agile Development.

★ Key benefits

☑ **Traceability ensures you deliver the product you intended.**
Each requirement will have a linked test case confirming that it was, in fact, included and that it matches the most current version of the requirement.

☑ **Traceability allows you to identify artifacts and work items that are no longer needed.**
A good traceability process makes it easy to identify artifacts such as design docs, development tasks, and test cases that are no longer included in the current product requirements, thus avoiding wasted effort and confusion.

☑ **Traceability makes it easier to see the impact of changes.**
By mapping specific versions of requirements to their artifacts users can ensure that when changes are made, each of the linked artifacts is flagged for impact analysis.

☑ **Traceability facilitates a robust change control process.**
The workflow can lock down requirements once they are linked to development artifacts, requiring a formal change control process to approve any updates.

☑ **Traceability makes testing more efficient.**
When tests cases are linked to requirements, they provide a roadmap for the testing team to know exactly what needs to be tested, and helps to focus tests on the required functionality and user experience, not just finding obvious bugs.

Summary

The lack of a Traceability process often leads to higher costs and delayed products. By linking underlying requirements to the planning, development, and testing processes, you are ensuring that your released products deliver what is intended and relevant.   An effective traceability process is the key to delivering product quality.

DevSuite's approach to traceability

DevSuite, from TechExcel, is a Product Development Lifecycle platform designed to enable comprehensive requirements traceability from product design through development, testing, bug fixing, and release. With an integrated architecture that facilitates dynamic linking of requirements to all subsequent development artifacts, DevSuite ultimately helps Product Management, Development, and QA teams to all work together to deliver the highest quality products.

For further information on DevSuite, please visit www.techexcel.com

Corporate Headquarters
3675 Mt. Diablo Blvd., Suite 200 Lafayette, CA 94549
Tel: (925) 871-3900
www.techexcel.com

TechExcel EMEA
Crown House, 72 Hammersmith Road London W14 8TH, UK
Tel: +44 (0)20 7470 5650
Email: emeainfo@techexcel.com