**Traceability is the Key**

Traceability is the key to building the right software, whether software is the product or just a part of the product. Traceability refers to the ability to define, capture, and follow the links between requirements and other artifacts, such as test cases, test runs, defects, and source code files, in the project.

The value of linking artifacts can't be overstated. Downstream artifacts originating from requirements can be discretely measured and evaluated for success or failure. If they are linked all the way back to requirements, then the successful evaluation of those artifacts means their corresponding requirements have been satisfied.

So, how does traceability work? When product requirements have been finalized and approved, testers write test cases to validate that the resulting software accurately meets those requirements. The test cases are executed one or (usually) more times. If a test case fails, a defect is recorded and routed to a developer to fix. After the developer fixes the source code, it is checked into the source code management system. The changed source code file is linked to the defect as part of the code check in process, which also marks the defect as ready for tester verification. With traceability links in place, the tester can quickly check to see which test cases need to be re-run. After the fix is verified, the defect is also marked as fixed. Because the defect was linked to a test execution result—which was linked to a test case, which was linked to a requirement—it's now clear to the project team that the requirement has been implemented and verified, and can be considered done.

Traceability’s true power is in enabling the team to quickly identify the impact of a defect or change request, easily address that impact, and keep the project on track.

Getting traceability right requires managing a lot of little things. Links need to be created and maintained across all of the product development artifacts. You need visibility into those links, and the ability to trace them upstream and downstream, to truly understand the impact a requirement has on the deliverable. Finally, when requirements change or business priorities shift, understanding the impact of the change needs to happen quickly.

Creating and maintaining links, without adding overhead to the project, is the key challenge to implementing effective traceability. In today’s lean organization, there is no downtime for the team when they can update spreadsheets or a homegrown traceability matrix. The time needed to manually compile a trace matrix or perform impact analysis just delays the project, making it that much more difficult and time-consuming to respond to change.

**Tracing Artifacts in TestTrack**

Tracing artifacts in TestTrack is both easy and seamless within any process. Let’s start with requirements, for example. You can create and manage requirements or user stories hierarchically, and organize them in TestTrack to define the product or component you’re building. Because requirements can be arranged hierarchically, they are automatically related to parent requirements. If a child requirement changes, or is noted as not met, this hierarchical relationship ensures the parent is also marked as suspect.

Parent-child relationships can be easily represented through the hierarchical structure of the document, and linking is automatic. But it’s also possible to relate requirements at the same level to each other, if a parent-child relationship isn’t appropriate. This action creates links that may be based on a functional relationship, rather than a hierarchical one.
Linking Test Cases and Requirements

As you generate test cases from approved requirements, those test cases are automatically linked to the requirement. This ensures a link from every requirement to the test cases that will validate the requirement. Even if you don’t automatically generate test cases from the requirements, you can write your own test cases and link them manually.

Every execution of a test case is also automatically linked to the test case itself, enabling testers to determine if requirements have been met based on the results of test runs. If each requirement has one or more test cases linked to it, and the test cases accurately reflect the intent of the requirement, successfully executing those test cases lets you confidently assert that the requirement has been met.

With all of those links in place, it’s easy to view the progress of your testing effort with traceability reports. You can see test execution results broken out by requirement, providing insight into the team’s progress in validating requirements. The traceability matrix also gives you the ability to look at test coverage, identifying requirements that don’t have associated test cases and prioritizing them with the team.

You Now Have the Key

Once you’ve linked requirements with downstream artifacts, you have the foundation of traceability in place. Traceability with TestTrack is a lightweight process that doesn’t require a lot of overhead. Because much of it is automatic, and creating a manual link requires only a mouse click or two, linking artifacts becomes a natural, and mostly invisible, part of the team’s daily work. The value of linking is that there is no question whether a specific requirement or user story has been met, and when it has been met.

By linking requirements to downstream artifacts, team members can be confident that the product they’re designing, building, and testing will meet all of the documented user needs.

We Live Quality

Quality is the way of life at Seapine. It’s in our DNA. From our expert Consulting teams, to our award-winning application lifecycle management (ALM) solutions, to our world-class customer support, Seapine has helped thousands of companies worldwide build, test, and deploy quality software. Now, let us help you live quality.

Visit www.seapine.com to learn more.