



Software Lifecycle Integration
A Quality Management Discipline

Table of Contents

Overview	1
Defects aren't the only shared artifact	2
Methods of sharing artifacts among systems	2
Using Tasktop Sync	3
The Quality Management System as a Dashboard	4

Overview

The testing team's job is difficult enough. Whether it's:

- Keeping up with changes in technology, such as moving from testing client-server applications, to testing cloud-based and mobile applications;
- Keeping up with advances in testing methods – for example, knowing when and how to use automated tests versus advocating for more exploratory testing;
- Having to cover all disciplines of testing, from functional and usability test to security and load testing - each requiring highly skilled professionals;
- Trying to establish standardized processes and reporting methods, in the face of project teams who think that each project is “special” and deserves an exemption; and
- Dealing with crushing timelines.

There was never enough time allocated to test before, and now Agile development methods seem to compress those timelines even further.

But in the face of all of this, the testing team really can be the organization that can create order out of chaos. The testing team is in a unique position to understand and assess the health of the project. Are all the requirements fulfilled? Are their implementations defect-free? Are the known defects tolerable if they're deployed into production? In many organizations, it's up to the testing team to declare whether an application is ready, or if they must stop-ship.

But in order to effectively discharge this responsibility, the testing team needs access to the right information. The hope was that the test management tools, such as HP Quality Center, could be the single source of truth, at least when it came to the managing the defects.

But development project managers have a different point of view on the same defects. They've brought in tools that help them triage and assign the bugs to developers to fix. And they're not using the test management system for that. They're using their own planning tools to manage defects.

Defects aren't the only shared artifact

As it turns out, each discipline in the software development and delivery process has their own point of view on the shared artifacts, and each of these disciplines have their own set of tools to create and manage these artifacts.

Testers need access to the requirements in order to create tests. But business analysts, product managers and product owners that write the requirements won't do that in the test management system. They prefer to use purpose-built tools for developing requirements and user stories.

Similarly, the Project Management Office uses purpose-built PPM tools. And the help desk uses service desk support software. And so on for each discipline in the software development and delivery lifecycle.

Each of these disciplines are linked together by their desire to create and deploy the highest quality application possible, yet their ability to work together is hampered by the fact that their tools do not work together. And when these teams don't work well together, projects fail.

Methods of sharing artifacts among systems

Getting these tools to work together can be accomplished several ways:

1. A manual process – such as logging into another team's tool, or emailing a colleague for the information.
2. Batch data transfer – exporting defects and requirements from one system and bulk importing them into another. Of course, artifacts that are loaded in this way have only static information; they are not continually updated with the most current status.
3. Automated integration of the systems - if these systems were integrated and the artifacts in them continually synchronized, everyone would have the most up-to-date information and practitioners could even collaborate with one another over these artifacts, all while remaining in their tool of choice.

Using Tasktop Sync

Tasktop Sync is an integration server that provides fully automated, real-time, enterprise-grade synchronization among the disparate tools used in software development organizations. This allows similar artifacts in multiple tools to be bi-directionally synchronized. Once a tool is connected to Tasktop Sync, that tool is also automatically integrated to other ALM tools that are also connected to Sync. Examples of these tools include: HP Quality Center, HP ALM, HP PPM; IBM Rational Team Concert (RTC), Bluemix, RequisitePro, Rational DOORS, DOORS Next Generation, Rational Quality Manager; Microsoft Team Foundation Server (TFS), Visual Studio Online; CA PPM; Planview Enterprise; Atlassian JIRA; Rally; VersionOne; iRise; Blueprint; Jama; Serena; Tricentis Tosca; Zendesk; ServiceNow; and others.

An example, HP Quality Center, Jama and JIRA

As mentioned earlier, testers create many of their test plans based on the requirements of the application they're testing. A product like HP QC can help, by automatically creating test plans from requirements entered into QC. But product owners, product managers and business analysts create user stories and requirements in tools built for that purpose, such as Jama.

Similarly, when the tester finds defects, that tester enters and manages it in HP QC. And when it comes time to fix the defect, project managers assign the defects to developers, and developers find these assignments, in tools such as JIRA.

Their tool silos separate the disciplines from one another.

A far better solution is one that Tasktop Sync provides, where the artifacts across a project team are bi-directionally synchronized.

With Sync, each system that maintains information about defects has their own up-to-date version of the data. If one practitioner makes an update to a defect (in the case of the tester, creating the defect. In the case of the project manager, assigning the defect. And in the case of the developer, fixing the defect), all of the other practitioners see the update, in real-time, in their tool of choice. In our example, defects in JIRA and HP QC are kept in sync, and the information flows between these systems in real-time.

The same is true for user stories and requirements. The user stories created in Jama are automatically created, and bi-directionally synchronized, as requirements in HP QC. Our testers can now use the automated features in QC that allow them to automatically create test plans

from the requirements in QC – the very requirements that were originally written as user stories in Jama.

Connections to other disciplines

While Tasktop Sync can certainly help drive the alignment between testers, developers, and business analysts closer together, it can also extend to other disciplines. For example, when help desk agents receive a complaint from an end-user they enter a trouble ticket into their service desk software (such as ServiceNow). Tasktop Sync can help escalate the tickets into defects. If desired, a ticket in ServiceNow can automatically create a defect in the test management system, or in the development team’s triaging system. These policies are easily configured in Sync.

Tasktop Sync also supports the Project Management Office, connecting their PPM tool into the rest of the development lifecycle.

Or, it can integrate your organization’s test management system, to those of outsourced testing and development teams, keeping everyone in sync on project status. In real-time, with no tedious exports to and from spreadsheets.

In this way all disciplines, business analysts, product managers, product owners, developers, architects, testers, system administrators, project managers, scrum masters and help desk professionals, are in alignment by sharing the same project artifacts. The entire software lifecycle is integrated; Tasktop Sync provides Software Lifecycle Integration.

The Quality Management System as a Dashboard

With the test management system synchronized to other systems, the testing team gains a centralized view on project health. Regardless of which discipline is creating or updating information, it can all be accessed in real-time through the quality management system.

Testers:

- Have access to project information in their test management system, rather than having to log into another product in order to find the information they need to continue their work.
- Have access to the most up-to-date information, eliminating costly rework due to inaccurate information.

- Are able to collaborate on artifacts by updating and augmenting them in their tool, knowing that their updates will be automatically reflected in all other systems syncing that artifact.
- Can better leverage the features in their tools when those features rely on artifacts created by other teams – such as automatically creating tests from requirements.

Testing teams and Managers:

- Gain better visibility into their projects by receiving reports that pull data from multiple systems, rather than having a siloed view of their project data.
- Get more collaborative, efficient teams. Practitioners no longer use ineffective email threads to communicate, they collaborate on common artifacts in their tool of choice.
- Achieve a strong Return on Investment - Each practitioner can save 20 minutes each day by automating manual processes. Using conservative labor costs, this results in an ROI of 20x the Sync license costs... each year.

Moreover, with a fully integrated ALM infrastructure, the Testing Center of Excellence can institute standardized processes and tools, without being undermined by the tools and processes of other organizations. The centralized test management system becomes the “single source of the truth,” providing test managers and executives the project-wide insight that allows more control and better decision making, ensuring that projects remain on-track to deliver high quality software, on time. Every time.