TEAM ID: PNT2022TMID30386

UTILIZATION OF TESTING TOOLS:

UAT In Agile Environment

The agile environment is more dynamic in nature. In an agile world, business users will be involved throughout the project sprints and the project would be enhanced based on the feedback loops from them.

At the beginning of the project, business users would be the key stakeholders to provide requirement thereby updating the product backlog. During the end of each sprint, business users would participate in the sprint demo and would be available for providing any feedback.

Moreover, a UAT phase would be planned before the sprint completion where the business users would do their validations.

The feedbacks which are received during sprint demo and sprint UAT, are collated and added back to the product backlog which is constantly reviewed and prioritized. Thus in an agile world, the business users are more close to the project and they evaluate the same for its use more frequently unlike the traditional waterfall projects.

UAT Team – Roles & Responsibilities

A Typical UAT organization would have the following Roles and responsibilities. The UAT team would be supported by the project manager, development and testing teams based on their needs.

Roles Responsibilities Deliverables

Business Programme Manager • Create and maintain Programme Delivery plan

• Review and Approve UAT Test Strategy and Plan

• Ensure the successful completion of the programme on schedule and budget

• Liaise with IT programme Manager and monitor the progress of the programme

- Work closely with business operations team and equip them for Day 1 operation
- Sign-off Business Requirement Document
- Review the e-learning course content
- Programme progress report
- Weekly status report

Software testing and validation are essential when implementing any system designed to process, store, visualize or manage data. Most of this testing (e.g., functional testing, regression testing, security testing, integration testing, etc.) is done by IT members and developers in the development environment with the intention to find bugs and ensure the software meets functional requirements that were developed collaboratively with the business. Since developers may not have a lot of knowledge or experience with company operations, there is always the chance that they will misinterpret the requirements. Developer testing is also unlikely to detect the scenario where requirements were poorly defined and do not effectively capture customer needs.

User acceptance testing (UAT) is the final type of testing done prior to roll out of the system. Unlike functional testing done by developers, UAT is done by business users who are familiar with both user and business requirements. In addition, UAT is conducted in a virtual environment that is configured to match the production environment so that users can test real-world cases with realistic data. UAT's purpose is not to find bugs, as those have hopefully been found and corrected in prior functional testing done by the technical team. Instead, UAT seeks to confirm that the software being delivered meets user needs in the context of the business environment it operates in.

Given their involvement in the day-to-day operations of the business, end users participating in UAT can help catch issues might have been missed in functional testing. No matter how thorough functional testing is, there are some business needs which can only be understood by end users of the software. Good UAT can be very effective in clarifying requirements, unearthing additional requirements that only become apparent once users see the system, and discovering issues in parts of the software that were not tested in the development environment.

While UAT's purpose is simple (verify the software works for business users prior to go-live), its implementation is something that companiesoften struggle with. Its position in the latter stages of the software development lifecycle (SDLC) means that there is often much impetus to get the finished product out the door which can cause companies to skimp on UAT. Many organizations come to appreciate UAT's importance only after being forced into costly software rework due to post-release software issues. In this blog, we will discuss the key issues that derail quality UAT efforts, and also provide an overview of several quality UAT management tools which can help optimize this important stage of testing.

UAT Manual Process Challenges

Given the pressure for system roll out in latter phases of the SDLC, UAT processes need to be as smooth, intuitive, and efficient as possible to lessen the temptation for companies to minimize or even skip UAT altogether. Many UAT difficulties are because of overreliance on time-consuming manual methods (e.g., using spreadsheets for collaboration and tracking) that make UAT processes inefficient and hard to manage. A few common UAT challenges that can be effectively addressed with a good test management tool include:

Poor end user engagement. Planning, executing, monitoring, and evaluating results of UAT all require iterative involvement from many different stakeholders and end users who are often globally dispersed. As these users have day jobs, UAT managers often struggle to get them involved in UAT activities. Many project managers try to track and manage UAT processes with excel spreadsheets (or even Microsoft Word), but this just imposes difficulties on both test managers and testers alike, as the manual processes involved in sharing and updating spreadsheets are asynchronous and inefficient. In addition, end users sometimes have little technical aptitude and find the testing process overly technical and tedious. To optimize UAT management, it’s important to engage and motivate key business users with a smooth UAT workflow. A testing-dedicated application should be utilized that is easy to use, intuitive, and has features that help to remove the overheadinvolved in UAT processes.

Test documentation challenges. Manual UAT processes require end users to document their every step during testing by pasting screenshot links into a document or spreadsheet. Cataloging defects for later reproduction in this manner can be time-consuming and frustrating for testers and equally frustrating for developers who receive this information and have difficulty reproducing the defects. The amount of back-and-forth communication required in such cases can be painful for all involved. Instead, a good test management solution will effortlessly document end user test execution, ensuring compliance with all internal and external quality standards and providing record-and-play evidence for easy defect reproduction. In addition, a good testing platform will make defect tracking easier with features that effectively eliminate the scenario where multiple defects are reported by different users about the same problem.

Test scheduling challenges. The UAT for a specific business process may involve multiple users. It can be challenging to coordinate a smooth and efficient handover of testing tasks between key users who are globally dispersed, resulting in delays in your UAT process. A good test management platform will eliminate bottlenecks by enabling project and testing managers to assign different steps in the UAT for a single business process to multiple users, with notifications allowing users to run the tests in an efficient, sequential workflow.

Test Management challenges. Test managers need to plan test coverage, track test results with each test run, develop defect reports, facilitate communication between developers and testers, make sure UAT is done on latest software version, etc. Trying to accomplish all this via spreadsheets and email will quickly become unmanageable for any large implementation. A good testing application will facilitate effective test management by allowing test planning, real-time reporting and monitoring of testing progress, central defect management, and traceability, software version tracking, and in-sync communicationbetween different testing stakeholders.

UAT Management Tool Overview

Let’s review some popular tools used by industry-leading organizations for UAT testing.

Atlassian Jira Software

Atlassian’s Jira Software is a popular Agile project management and issue tracking tool that is used for software development. Jira is a platform independent tool that is compatible with many different operating systems and browsers and also has the flexibility to be licensed for running on-premise or as a hosted application. Featuring configurable Scrum and Kanban boards, Jira Software can capture requirements, create user stories, facilitate communication, track issues, create custom workflows (or utilize out-of-the-box ones), plan sprints, distribute tasks across the software team, create reports, and much more.

While Jira Software was not designed specifically for test case management, it allows customization of native components (i.e., issue type, workflows, etc.) to meet your test management needs. In addition, Jira software test management functionality can be extended by integrating it with a variety of common testing tools and frameworks.

Zephar Application

Zephyr for Jira is a native application that exists within Jira, which can deliver high quality test management capabilities to any Jira software development project. When Zephar is used with Jira, tests can be created and viewed in any Jira project, with tests executed either immediately or as part of a testing cycle that is linked to other issues. Detailed testing metrics can also be tracked with customizable Zephyr gadgets. Important functionality within the Zephar for Jira tool includes:

Create, view, modify, and execute tests

Assign test executions

Create a test cycle and add tests to test cycles

View testing progress via customizable metrics dashboards

One-click integration with automated testing frameworks – Jenkins CI, Selenium, Appium, Cucumber, and SmartBear products

Machine learning algorithms can help maximize automation by optimizing test case implementations and predicting defect trends

Zephar allows Jira users to better manage software releases by enabling access to testing data from within the Jira platform. This allows project managers to evaluate whether the software quality suffices for launch without having to go searching for data across other tools.

With a clustered horizontal architecture, Zephyr scales to meet the needs of every organization, providing a full range of support plans to accommodate teams of any size and any type of location dynamics. In addition to Zephar for Jira, Zephar Enterprise is available as a standalone test management solution outside the Atlassian ecosystem, with real-time Jira integrations that help to streamline your workflows.

Rally Software

Rally Software® is an enterprise class Application Lifecycle Management (ALM) Software-as-a-Service (SaaS) platform that serves as a project management and collaboration tool for Agile software development teams. The platform is a Jira competitor that is built for coordinating work across multiple teams and programs with large numbers of users and artifacts. Key features of Rally Software® include:

Real-time status/progress tracking with dashboards and burndown charts

Customizable dashboards (e.g., Scrum, Kanban, or custom views)

Iteration and release planning

Capacity and resource planning

Risk identification and management capabilities

KPI tracking and reporting

Manage QA and testing activities

Central hub with threaded chat and a team inbox that aggregates email, feeds, and tool notifications all in one place.

Integrates with other popular apps and tools (e.g., Jira)

Rally has several important features to help facilitate test management including the ability to create test plans and custom test plan lists, create and manage user stories, create test cases and associate them with a particular iteration, execute test cases, log and track defects, generate and manage test sets, generate test result reports, import or upload bulk test cases, and create mapping between all these items. With user stories, test.

Tricentis qTest Platform

qTest is a leading test management platform that supports the latest processes and methodologies such as Agile, DevOps, and BDD (behavior-driven development). The platform is comprised six different tools:

qTest Manager– A test case management solution that allows you to track, organize, and report on all testing related activities through an easy to use interface, with built-in workflows for teams that release software less often than once a week.

qTest Explorer– Manual testing tool (exploratory and scripted) that tracks all interactions of a testing session and automatically creates detailed test scripts and defect documentation.

qTest Scenario– A Jira app for enterprise teams practicing scale test-driven development (TDD), behavior-driven development (BDD) and acceptance test-driven development (ATDD) that facilitates test scenario creation and ensures future traceability to Jira issues.

qTest Pulse– An advanced rules engine that automates tool communication within a DevOps pipeline with built-in workflows for teams that release software continuously (at least once a week).

qTest Insights– Business Intelligence and data analytics engine, providing a central hub for customized reporting on all testing activities across the platform with shared dashboards.

qTest Launch – A central test automation interface that allows Agile and DevOps teams to manage, schedule, kick off, and report on automated tests across the organization.

The qTest platform provides a scalable and flexible testing suite with an enterprise architecture that is designed to cover use cases for organizations, not just individual workgroups. The platform integrates with issue trackers, development tools (e.g., Jira Software, Rally Software), test automation tools, and more.

Conclusion

Any software project will fail to provide the desired business value if the delivered software does not meet user expectations. User acceptance testing often uncovers missed issues and helps to clarify requirements for software applications. As such, UAT is critical to ensuring a successful system roll out.

Unfortunately, many organizations rush UAT processes, or even skip them altogether due to pressure to get the software out the door and into production. This problem is amplified because of overreliance on time-consuming manual processes that make UAT inefficient and hard to manage. Another factor that contributes to organizations skimping on UAT is reliance on legacy testing platforms that were designed to enable manual processes and/or do not support more modern Agile development approaches.

In this blog, we discussed challenges incurred when using manual processes for UAT and summarized several popular test management solutions that support modern development methods. When choosing a test management tool for your organization, key factors to consider include:

Supports both UAT and technical (e.g., functional) testing

Integrates with your existing tools

Supports your browsers and operating solutions

Easy for both business and technical testers to use

Offers good customer support

The bottom line is that if UAT is conducted properly by experts with deep knowledge of the project and its objectives, it will help to decrease rework costs and ensure user adoption. And an important part of any successful UAT is a good test management tool.