



The Science of Running Effective User Acceptance Testing Cycles

WHITEPAPER

User Acceptance Test (UAT) programs have traditionally been areas of contention between IT and the Business. IT teams get critical systems readied through development and testing, while Business teams verify that these systems meet their requirements. The division of responsibilities might seem clear cut, but realities on the ground are far different. The planning, management, execution and reporting phases of UAT cycles have always been problematic since responsibilities fall in grey areas.

This whitepaper will outline a well-defined 5-Phase UAT process that assists with Planning, Coverage, Execution & Tracking, Reporting, and Reuse.

Table of contents:

- 3** Planning
- 5** Coverage
- 7** Execution & Tracking
- 9** Reporting
- 10** Reuse



Planning

An important truth for any process where quality must be determined is the fact that the earlier an issue is found, the less expensive it is. This includes UAT as well. Traditionally, in waterfall methodologies, UAT doesn't occur until later in the cycle closer to the delivery date. Even today, this practice exists by default within organizations. The risk with this approach is simple: wait until the end game to discover that the requested functionality was misunderstood by development teams and the costs for fixing before release becomes much higher, risk is accentuated, and quality is often sacrificed.

UAT should occur early and often. Regardless of process, the planning phase for UAT needs to be included in the master project plan and scheduled across the life of the project. This needn't be complex or difficult to achieve - nor does the actual testing process need to take a long time. For example, XYZ Company practices Agile and schedules UAT every two sprints. It lasts for 1/2 a day or less, and provides feedback soon enough for the process to flex with changes. Company ABC takes a different approach and has a business representative participate in the agile team who verifies user acceptance each sprint within the agile process. UAT is scheduled every three sprints and is conducted by a team of analysts as well as two external customer representatives in order to assure the project remains on track.

Environments are another key factor in UAT testing: Where is this test going to occur? Steer clear of utilizing development and QA environments – these change frequently and any disruption in the development process could stress the schedule. Ideally, a separate environment for UAT would work best although a shared 'demo' environment can be used as well as an unused staging environment. Regardless of where the testing will take place it is crucial to begin setting this up well in advance of the UAT to be sure everything is working. Finally, this goes without saying: Always, always, always have QA do a full smoke test on the UAT environment at least a day prior to the customer's access. This will give time to correct any inconsistencies that may be found.

Planning Checklist:

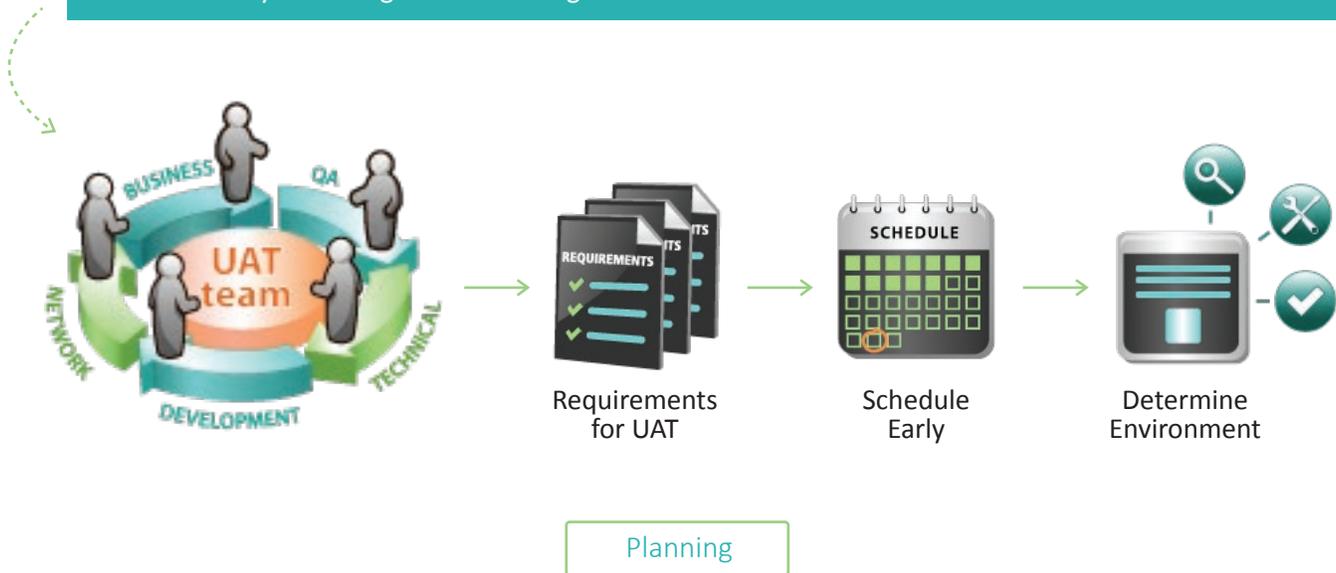
- 1** Determine the point person for directing UAT testing. This can be someone from the business, a technical project manager, a QA lead, etc.
- 2** Form a UAT team. A mix that works well is a combination of all three from the above (or more).
- 3** Collect the requirements that will be verified. If using an Agile process, the Acceptance Criteria usually provide the necessary information.
- 4** Assure UAT testing falls at an earlier point on the project schedule. This does not negate a final UAT which is often required by the customer.
- 5** Identify, build, and verify the environment well before the testing date.





How Zephyr can help with Planning:

Zephyr's Test Repository and Scheduling applications make it very easy to identify areas of test and assign and schedule those specific tests in UAT cycles. The Business may or may not be involved in the direct creation of test cases in this phase. If UAT checklists or tests already exist, they can be easily imported into Zephyr and tracked. The planning and scheduling process is real-time and is visible to both teams. User acceptance tests are very easily authored in the Test Case Creation application that has an Excel-like feel to it, thereby allowing non-IT resources to easily interact with it – be it for actually creating tests or merely reviewing and annotating them.



Coverage

User Acceptance Testing is often confused with a 'regression by client.' This often occurs because expectations haven't been clearly understood or communicated throughout those involved in the process (including the customer). UAT is defined as the process whereby the customer verifies requirements that have been requested exist and provide the functionality as outlined in the user story or requirements document. Hence the term 'Acceptance.'

UAT should not be considered to be a functional regression of the software or a time to change requirements and log the changes as bugs. However, always remember: Bugs will likely be uncovered, especially if the UAT falls earlier on the schedule, and the customer often has a better idea of what they really want once they see the functionality. For any project it is important to mitigate both of these challenges. It is best to have a QA lead involved in the UAT process as a point person to receive any issues that are found in the software. The lead can then review the list for known issues and add any new ones as bugs. Managing requirement changes is a function of the project team – they determine what the process is for those changes. If using Agile, a common strategy is to introduce the stories into a sprint where appropriate and negotiate schedule overruns with the customer.

Given the above, it is important to clearly define the requirements that will be reviewed by the customer. If using Agile, either the stories or the acceptance criteria will provide the necessary information. In other processes, any document or system which identifies and tracks requirements against development will provide the required coverage specifics.

It is also helpful to have a set of test cases to track against during testing. These don't need to be full blown, step-by-step instructions in most cases but they are useful for encapsulating the requirements and directing folks through the application. Also, using a test management system to track the execution and progression of testing can be a valuable exercise both during the testing and as a historical document to refer to, should questions arise in the future around what exactly was looked at.

Coverage Checklist:

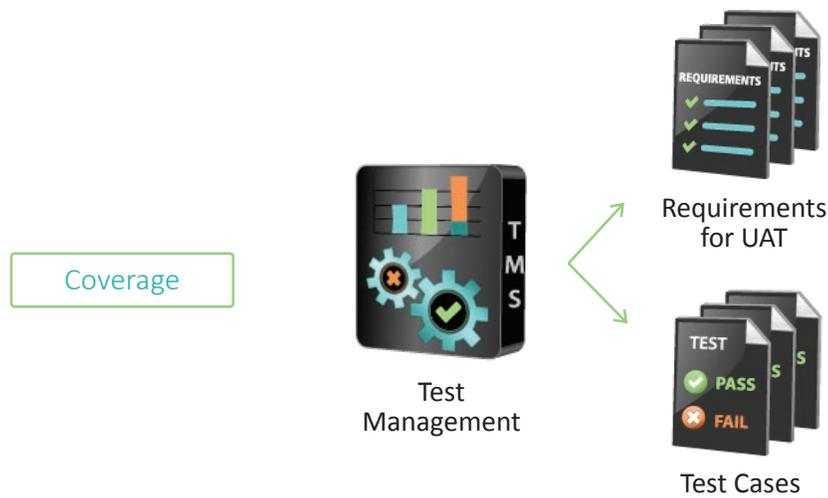
- 1 Set clear expectations around desired outcome for UAT testing.
- 2 Involve QA to assist with capturing issues.
- 3 Identify the requirements to be reviewed. Stick to these only!
- 4 Capture requirement changes and fit into project as warranted while keeping an eye on project slippage and delivery schedules.
- 5 Use a system (such as Jira, GreenHopper, or a robust test management system) to track requirements under test.
- 6 Write test cases which encapsulate the requirements that need to be reviewed. These need not be complicated or indepth for most UAT testing however, be sure to provide enough information so the testers, whether they be internal or external customers, can navigate and understand what is being asked.





How Zephyr can help with Coverage:

Test Coverage of the various features and functionalities that have been added to the new systems is very important. The Business needs to get a good feel - through both quantitative and qualitative means – that the system indeed does meet their requirements. Zephyr's Requirements Management application in conjunction with the ability to directly map UAT testcase(s) to those requirements allows for real-time coverage reports and tracking metrics. Very simplified click-n-select processes allow for such mapping. End-to-End Requirements traceability is also available.



Execution & Tracking

There are several ways to perform User Acceptance Testing: Hands on in-house (lab, training room etc.) and remote. In some cases, if there is no way for the customer to interact with the application, an in-depth walk-through often suffices. The execution phase often holds the most anxiety of the UAT process. What if the customer doesn't like it? What if there is a major issue? And so it goes – that nagging fear of software development. This is the phase within which to relax. If you're not prepared for the UAT and have questions as to whether things will go wrong then cancel it! The execution phase is an opportunity to get closer to your client – whether external or internal. Full disclosure is important in UAT. If there is a major bug, don't hide it – discuss it and work around it. Issues during the software development process are normal and expected and helping the client understand this will go a long way in building a trusting, positive relationship.

The team for UAT testing should include at minimum a business representative (Product Owner etc), Technical Project Manager, and a Quality Assurance representative (preferably the lead QA engineer for the project). It is also recommended to have a developer and network admin on-call for unintended occurrences (and anyone else you identify as important). You will know the best make-up for your team given your process, product, etc.

As mentioned earlier, it is helpful to have basic test cases written and available in a test management system that tracks execution. They should be organized under UAT and kept separate from on-going project work. The customer can either interact with the test management system or the execution can be tracked by a member of the UAT team. Regardless, a historical record of test case executions should be recorded and saved. This does not negate the importance of allowing customers to interact with the system in an organic natural way (black box testing) but rather, provides a framework within which to establish the guidelines for the UAT testing to assure all requirements have been reviewed. Always remember: Clear expectations are key for a successful UAT.

UAT can be performed from anywhere. Customers can visit and use local machines or their laptops, the application can be made available over the internet, VPN privileges can be given, etc. The best place for UAT is determined by multiple factors and must be addressed by the project team. Regardless of test location, all of the recommendations in this paper can be utilized. Finally, assure you have the full support of your networking team prior to any UAT execution – especially if you are performing it remotely.

Execution and Tracking Checklist:

- 1** Determine the location for testing: in house, internet, VPN etc. and assure network support is available prior to testing starting.
- 2** Assure that you are prepared for the customer to view the application. Bugs are OK but be sure the client is aware.
- 3** Use UAT to build relationship and trust with the client.
- 4** Define a dedicated SQA role and either have them present at the event or participating remotely. They will be invaluable for clearing up questions regarding issues, functionality, etc.



- 5 Assemble a technology team who can be on-call to support the event. A development and network engineer are a good start.
- 6 Track test case executions or the areas being looked at and record all results.

How Zephyr can help with Execution and Tracking:

UAT execution in Zephyr is highly simplified to allow non-technical domain experts and business users to easily run them. Akin to performing simple everyday tasks such as using a Web Browser or Excel spreadsheets, the Test Case Execution application in Zephyr is accessed via a web browser. The user logs in with a username/password provided to them, clicks on the Test Case Execution application and clicks on the UAT Cycle. A list of user acceptance tests is presented to them. On selecting one of them, the detailed steps to perform are shown. The user can then (with a simple drop-down menu) “pass” or “fail” test steps and/or entire tests and type in their comments. They then move on to the next test. That’s it. They don’t have to fill out any spreadsheets, or keep track of where they left off the execution or write reports.



Reporting

Two underlying themes throughout this paper have been communication and setting expectations. A successful UAT does not ignore the Reporting phase if it seeks to assure that the customer is happy and everyone has agreed with the results of the testing. The Reporting phase encapsulates and communicates the degree to which expectations were met. It also provides a historical document from which to mitigate future questions or issues that may arise. The report will need contributions from the business as well as SQA and any other project participants deemed necessary.

The UAT report can include as little or as much information that it needs in order to fulfill expectations. A report for an external client may consist of more information than that of one used internally because internal systems hold much of the historical information. However, regardless of client orientation, it is advisable to always document the results.

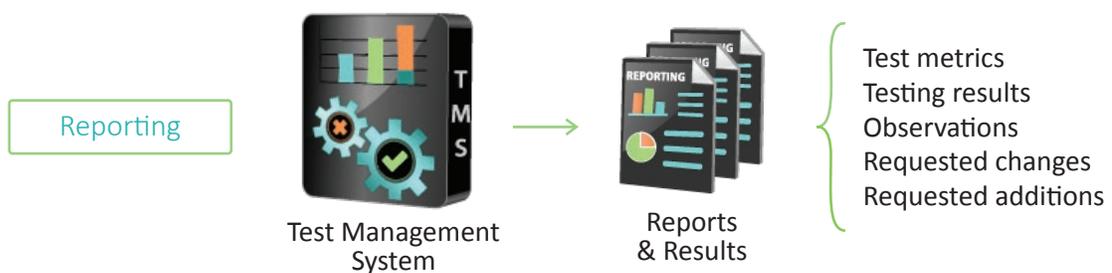
Recommended content for the UAT Report:

- 1 Overview
- 2 Requirements/stories under test
- 3 Test cases
- 4 Test Metrics - # tests executed, #passed, #failed, pass/fail percentage, who executed tests
- 5 Testing results
- 6 Free-form testing results (black box testing)
- 7 Observations
- 8 Requested changes
- 9 Requested additions



How Zephyr can help with Reporting:

The Zephyr systems monitors all test activity in real-time and presents the rest of the project team and upper management with this information in live Dashboards. Live Dashboards show details about the UAT execution schedule, project news and detailed metrics about how well the UAT cycle is progressing in real-time. They show progress, pass and failure rates, areas of risk, defects and provide overall quality insight. All of this without having to build reports, run queries, hound the non-IT users for updates or do a ton of paperwork.



Reuse

Most projects will require more than one UAT, especially if the ‘test early and often’ approach is used. Remember, the earlier an issue is found in any project, the less expensive it will be to fix. This holds especially true when UAT issues are discovered late in the process close to the delivery date. When subsequent UAT tests are performed, it is best to test previously reviewed functionality as well. To maintain consistency, it is advised to utilize the same test cases that were executed in previous UAT iterations. Not only will this reverify customer acceptance, but it also builds trust by showing that a consistent quality system and process is in place. It is advised that a system which allows a new iteration to be created and run is used so results are stored separately from previous sessions and customers see unexecuted test cases.

Reuse Checklist:

- 1 Use a system which will allow new test iterations to be created with previously executed test cases
- 2 Re-test previously tested functionality as well as new
- 3 Build confidence and trust with customers by presenting an organized test process and system

How Zephyr can help with Reuse:

Once the UAT cycle is completed successfully, none of this information is thrown away but is recycled for the next UAT cycle. Tests, schedules, reports, metrics are all reused and modified appropriately for a new cycle. This tremendously speeds up successive UAT cycle planning and execution. All historical information about past UAT cycles is stored and available immediately for instant comparisons and reuse.





User acceptance testing is a valuable process for any project. Whether it is an internal or external facing application, it is important to verify that the application meets the expectations of the end user. Contention between the business and development organizations need not exist and can be mitigated by advanced planning, setting expectations, and open, clear, concise communication. Forming a small team of individuals from various departments will create ownership across the organization and, if executed well, spread the burden of the UAT event among multiple individuals. UAT need not be a negative, time consuming process.

Using systems and simple process will add to the ease of UAT execution. Storing and executing predetermined test cases from a test management system not only lends a core stability to the process, it also demonstrates a high degree of organization and consistency around the quality process; something the customer will notice and appreciate. After all, UAT is all about the customer. Reframe UAT from a 'must do' process to an opportunity to further engage the customer in a positive, trust building relationship. As outlined in the previous pages, UAT is not complicated and doesn't need to take a lot of time to prepare for. Start early, form the correct team, outline and schedule tasks, set clear expectations around coverage, utilize test management and process technology to provide a solid core for the project, track results, and provide a full, detailed report. All that remains is engaging the customer in high integrity communication throughout the process and using the experience to build a stronger, more trusting relationship.

About Zephyr

Zephyr is a leading provider of quality management solutions, powering intelligent DevTestOps for more than 11,000 global customers across 100 countries. Project teams and enterprises of all sizes use Zephyr's products to enable continuous testing throughout their entire software delivery pipeline to release higher quality software, faster. Zephyr is headquartered in San Jose, CA with offices in King of Prussia, PA, Europe and India. For more information, please visit www.getzephyr.com.



Contact Zephyr Today!

sales@getzephyr.com

www.getzephyr.com

+1-510-400-8656