

UAT execution and report submission

application development has to meet its requirements. As well as ensuring that these are all identified and understood, a Requirements SME needs to specify the acceptance criteria in terms that can be understood by the entire team. As soon as an issue or defect is identified, it is fixed or resolved before the product is deployed, thereby saving time and cost.

The Warning Signs

Concentrating on other types of testing such as unit testing, integration testing or performance tests at the expense of UAT because of project time constraints

If testing focuses on the technical functionality of the software instead of looking at the complete end-to-end testing process, defects will only be discovered after the product has shifted to production, and will be likely to cost at least ten times to fix than had it been captured earlier in the process

Domain experts and business staff who lack the know-how and cannot accurately identify scenarios relating to the operation of the whole business process which needed to be tested.

With no expert input to extensively prepare and manage a user acceptance testing process and risk assessment errors will not be captured before the system goes live, the software application will not behave as the end users anticipate and the developers will be swamped by change requests from the business at the worst possible time to find and resolve major defects.

Requirements Subject Matter Experts must understand business and application software requirements and critical business flows, and be able to competently prepare and validate data which are realistic and pertinent to the business.

UAT is seen as part of the deployment process rather than development

UAT should be an integral part of the development process so that development teams can get immediate warning of any misunderstanding of the acceptance criteria. This allows teams to make changes or a modification to the criteria at a point in the application lifecycle where it is easy to do so.

There is no clear UAT plan in place

An effective UAT plan needs to detail out the key focus areas, do a risk assessment and prepare the user acceptance test cases to test the system. These should use real world test scenarios for the applications and be in sufficient detail to implement scenario-based testing to conduct the UAT process.

There is no Change Control process

Change control is as important as issue tracking in development. It makes all changes visible to the team and makes sure that team members can bring their expertise to bear on decisions on any proposed changes. This provides a better understanding of their impact and risks are understood. If Change control interferes with progress, it means that there are too many changes. Complaining about change control is like complaining about the loud noise of a fire alarm when the building is in flames.

UAT defect management is ad-hoc

There must be an effective defect tracking and management process that allows the delivery team to review test progress and metrics with the testing team, determine priorities and to resolve all defects found during UAT testing. All high-priority defects that would prevent a deployment can be quickly.

Testing is done in isolation by specialized testers

UAT involves testers, developers, business representatives, Governance, administrators and third parties. It is an aspect of development. If, for example, compliance experts aren't involved, then it is easy to miss a statutory requirement that affects the basic test scenarios. If business representatives aren't involved, then obvious defects can easily be missed. All team members that are involved in the UAT process need a clear understanding of their responsibility, roles and tasks. All test needs and models must be discussed with the developers, QA team representatives, database administrators, Operations staff and the testing team.

Ambiguity in the definitions of requirements

Ambiguous requirements often leads to testers and developers to unknowingly misinterpret them, which causes requirement errors to slip through UAT and morph rapidly into defects. These defects surface after implementation, at a time when it costs tenfold to pursue and resolve the same defect. It is therefore important to capture and define these shifting, ambiguous requirements and flesh out

meticulous acceptance criteria so that the intended results are visible to all. Ambiguity can be one of a range of critters. These include

Ambiguous terms : subjective or vague terms that cannot be measured

Conflicting requirements : Two or requirements that conflict each other

Incomplete requirements : Missing values, business rules, etc.

Missing requirements : Possible missing requirements that have not been defined

Unclear requirements : Requirements that can be interpreted in multiple ways

Glossary : Term is not found in the glossary reference document

Grammar, spelling and wording : Spelling mistake, grammar, rewording suggestion

Key Steps When Implementing an End-to-End UAT Process

User Acceptance Testing is a challenging part of software delivery. If the process is fudged, it is too easy for buried issues to materialize quickly and escalate to the point that they risk the successful delivery of the development project. Because software delivery relies on being able to change rapidly in response to changes in the business requirements and a better understanding of the business domain, UAT must rapidly change its criteria and scenarios to remain in step. This means that it is a process with its own lifecycle. It can be agile, but it must never lose precision. We will discuss the process in a series of sections.