

BA AND UAT

The Business Analyst (BA) role is focused on understanding what the business needs and ensuring that our projects deliver on those needs. And while the solution requirements are a natural outcome of the BA's work, that "front-end" work is only the beginning of the ways the BA can ensure that our projects deliver business value.

Requirements specify the "what" but not the "how." While they clarify what must be delivered (both functionality and qualitative attributes) to meet the business needs, requirements will always leave room for the implementation team to choose how to implement the solution. It is proper for the implementation team to make these choices based on technical imperatives and recognized best practice. But these choices can sometimes impact upon the users of the solution.

User Acceptance Testing (UAT) provides an important check to be sure that the business needs have been fully and correctly met, and that they have been met in a way that is acceptable to the business and the users. Participation in UAT provides the BA with a significant opportunity to ensure that the solution provides precisely what the business needs.

Let's explore the ways in which a BA can participate in UAT in order to ensure value delivery.

What is User Acceptance Testing?

UAT (User Acceptance Testing) is unique among the activities we undertake on our projects, because its focus is on people who are outside the team, and its primary purpose is not finding defects.

User -- The people who will actually use the solution are not generally active in the project that creates it. This is true even in Agile projects, where the Product Owner (who is quite involved) is only one person, and the other users remain more distant. UAT gives users the opportunity to see what has been built and provide feedback to the development team before it is deployed.

Acceptance -- No system is perfect, so we must ensure that our system is acceptable before it is deployed. And it is precisely the Customer for whom the system is being built and the ultimate end users (those people who are not very involved in the project most of the time) who must accept what we deploy. UAT is their opportunity to determine if they can indeed accept what has been built.

Testing -- Written requirements, drawings, mock-ups and the like are not the system, they are merely attempts to clarify what the system will be and do ahead of time. When it comes to acceptance, those things, and even demonstrations of the

system, are no substitute for the Customer and the Users actually putting their hands on the system and trying it out (testing it).

User Acceptance Testing is when the people who will ultimately use the system to do real work run it through its paces in a test environment to be sure that it meets the business need well enough to be accepted.

Identifying UAT Testers

The Business Analyst (BA) is in the perfect position to identify the people who should be the UAT testers. By virtue of the Requirements Elicitation and Analysis activities, the BA has already identified and made contact with a wide variety of end users; both Active and Passive users.

The Active Users are those who actually interact with the system itself. These are the people whose hands are on the keyboard (or mouse or touchscreen or . . .) and would have been key sources not only of functional requirements but also of performance, usability and other non-functional requirements as well. Much of what is built into the system is intended to meet these people's needs, so some (or many) of them should definitely be UAT testers.

The Passive Users don't interact with the system directly, but are impacted by the system nonetheless. For example, they may receive reports, emails or notifications from the system, or their jobs may include using the information the system produces, or they may supervise Active or Passive Users. The acceptability of the system for these people's purposes is no less important than for the Active Users, so some of these people should be UAT testers as well.

While identifying UAT testers, we need to think not only about the end users we identified during Requirements Elicitation and Analysis, but also about the requirements themselves. User Acceptance Test should include checking *all* of the functionality of the system for acceptability, so it is useful to go through the requirements and ensure that there is a user who will check each one during UAT.

Planning UAT Tests

Test Planning for UAT is different from planning the other types of testing on our projects because its purpose is different. All of the other testing we do is focused on finding defects, so by the time we get to UAT there should be few defects remaining to be found. The focus of UAT should be on how the system will actually be used in practice.

There are two quite common shortcuts to UAT planning that we must avoid:

1. The first is to simply reuse the tests that our QA Testers wrote. That is wasteful of the UAT testers' time because those tests have already been run

and passed. It is also inappropriate because those tests are focused on finding defects (the purpose of system testing), not judging acceptability (UAT's purpose).

2. The other mistake is to not plan UAT tests at all -- to just tell the UAT Testers to try the system out. Since UAT testers are rarely testers by training, this sets them up for not achieving the purpose of UAT.

Because the Business Analyst learned about the business needs and business activities during Requirements Elicitation and Analysis, he/she has an understanding of what should be tested during UAT and should be involved in planning the UAT tests. UAT planning should go something like this:

1. Create a separate test plan for each type of User who will participate in UAT (both the Active and the Passive users). The test plan for each type of User must include each business scenario that role will use the system to work through.
2. Be sure to include not just the normal "happy path" scenarios, but also all of the special cases and error cases. (Error cases would include the user making a mistake, the user encountering mistakes other people have made, and things that can go wrong with the system or other systems.)
3. Give the tester appropriate instructions for testing each business scenario. "Appropriate" instructions ensure that the tester tests all of the things he or she should test, but don't include so much detail that the tester could mindlessly follow them without truly evaluating acceptability of the system.
 - For the new functionality of the system, the instructions should include step-by-step how-to, and should prompt the tester to evaluate if that way of doing the function is good from a business usage perspective.
 - For existing functionality, the instructions should just tell the tester what to do (prompting the tester to use his or her existing knowledge about the system and the business process).
 - In any test where the data is important to the test (e.g. for special or error scenarios), the instructions should be precise about the data that the tester must use. In other cases, choice of data should be left to the tester.
 - In all cases, the tester should be prompted to check these things:
 - That the system operates as required and handles data and computations correctly. (Defects may have escaped detection in prior testing.)
 - That the way the system operates and presents information will work with how the user's business processes flow.
 - That the usability, performance, security and other non-functional attributes are appropriate to the business scenario.
4. Go through all of the requirements and ensure that each will be tested by all of the users that it is designed for. Add to the individual user role test plans as appropriate to achieve complete UAT coverage of each requirement.

5. Note that it is fine (even appropriate) for more than one UAT tester to test the same functionality if multiple user roles use that functionality. This is especially important if the different user roles use the functionality in different ways or for different business scenarios.

If testing specialists are available, the BA should collaborate with them to assure that good UAT plans are prepared. Good User Acceptance Testing requires appropriate test strategies (something a testing specialist will contribute) married with an intimate understanding of the business needs and constraints (which the BA brings to this collaboration).

Determining When To Do UAT

We usually think of UAT as being the final step of the project immediately before deployment. But doing this important evaluation so late in the project represents significant risk -- any issues found in UAT will either cause delays (as they are corrected), or will have to be accepted (and possibly scheduled for correction at some future date).

For this reason, the BA should work with the developers and testers to identify ways that UAT can be done earlier so the users' feedback on the system can be incorporated into the software without endangering the project timeline or costs. Here are some examples of ways to do this:

- In **Agile** methods (like Scrum), the software is developed in small Sprints of only a few weeks. This provides the option of doing mini-UATs (involving appropriate users) after certain Sprints to evaluate chosen functionality.
- Traditional (non-Agile) projects are often broken into **Phases** of a few months each, with certain functionality being built in each phase. If this approach is being used, each project Phase should end with UAT.
- Even when the project is not broken into Phases or Sprints, a **Function-at-a-Time** development approach would allow UAT of each function as it is completed.
- A variation of the Function-at-a-Time development approach is the **Walking Skeleton** approach in which the skeleton of the software is built first. (The skeleton would include the product's structure, menus, and placeholders for each window, screen, and dialog.) After UAT of the skeleton, the development team adds a function at a time to the skeleton, allowing Function-at-a-Time UAT.
- Even if none of these development approaches is being used, it is still valuable for users to be given the opportunity to review screens, menus, computations, reports and other things as they are developed.

Doing UAT early and incrementally is your best insurance against unhappy surprises at the end of the project; so working with the development team to find ways of doing this is effort well spent.

Delivering Business Value Through UAT

The BA role is all about ensuring that each project delivers the value the business needs and expects. Actively participating in UAT planning and execution is an important way for the BA to ensure that value is indeed delivered.

Upcoming Webinar

I presented a webinar on this topic in April 2016. You can [listen to the recording here](#).

Posted at 12:21 PM | [Permalink](#)