Overview of User Acceptance Testing (UAT) for Business Analysts (BAs)

by Vincent J. Bordo, CBAP
January, 2019

User Acceptance Testing (UAT) is one sure way to reduce or eliminate change requests, and drastically reduce project costs. If your organization does not practice UAT or does not have a mature process of UAT, this article provides information to hopefully persuade you to re-consider. UAT is an effective process with a high rate of return for those who take the time to implement and follow its discipline. If missing from the software development life cycle, your organization is missing a great opportunity to improve project success.

Testing accomplishes a variety of things, but most importantly it measures the quality of the application. Testing presupposes there are defects in the software waiting to be discovered and this view is rarely disproved or even disputed. Several factors contribute to the importance of making UAT testing a high priority of any software development effort; these include:

- Reducing the cost of developing the application. Minimal savings that might occur in the early stages of the development cycle by delaying testing efforts are almost certainly bound to increase development costs later.

- Ensuring that the application behaves exactly as expected. For the vast majority of programs, unpredictability is the least desirable consequence of using an application.

- Reducing the total cost of ownership. By providing software that looks and behaves as shown in your documentation, your customers require fewer hours of training and less support from product experts.

- Developing loyalty and word-of-mouth market share. Finding success with a program that offers the kind of quality that only thorough testing can provide is much easier than trying to build a customer base on buggy and defect-riddled code.

What is User Acceptance Testing?

User Acceptance Testing (UAT) - also called beta testing, application testing, and/or end user testing - is a phase of software development in which the software is tested in the "real world" by the intended audience or a business representative. Whilst the technical testing of IT systems is a highly professional and exhaustive process, testing of business functionality is an entirely different proposition.

The Goal of UAT Testing

If you ask somebody the question “What is the goal of a software test?'' you might get an answer like: “The goal is to prove that a system does what it is supposed to do''. This answer is not exactly correct and demonstrates the necessity to define some fundamentals about software testing. Another response would be, “The goal is to find faults or defects''.

The goal of User Acceptance Testing is to assess if the system can support day-to-day business and user scenarios and ensure the system is sufficient and correct for business usage.

Where does Testing Fit In?

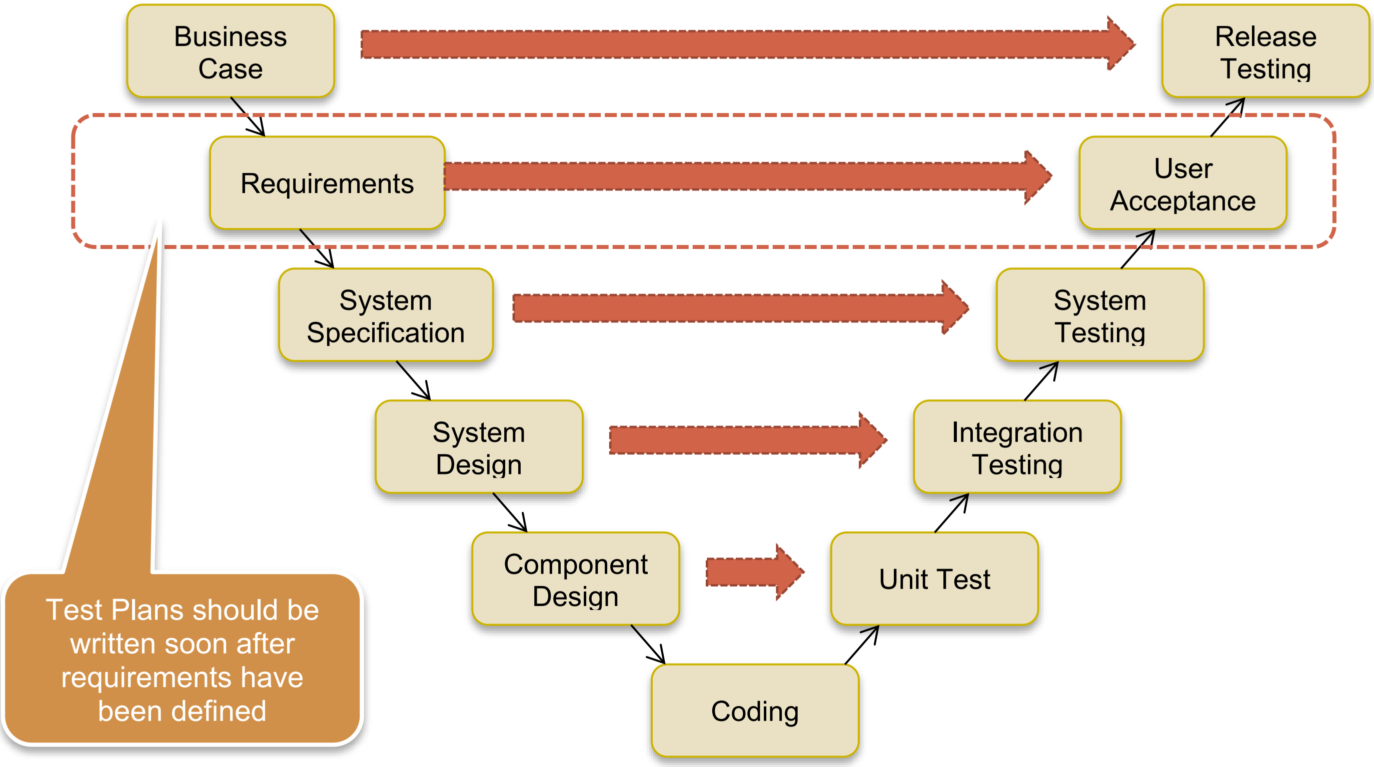
When a software developer writes code, it is common for mistakes to occur, such that requirements are not adequately implemented or they are forgotten. It is during this process that errors are introduced into the system. It is also possible that the business had not communicated their requirements correctly, or they could have insufficient details, which could result in a system working as designed, but not as expected.

UAT tests are created to verify the system's behavior is consistent with the requirements. These tests will reveal defects within the system. The work associated with UAT begins after requirements are written and continues through the final stage of testing before the client/user accepts the new system.

The "V" Model for Testing

The "V" model is a methodology where development and testing takes place at the same time with the same kind of information available to both teams. It is good practice to write the UAT test plan immediately after the requirements have been finalized.

The "V" model shows development phases on the left hand side and testing phases on the right hand side.



Why is Testing Important?

Most of us have had an experience with software that did not work as expected. Software that doesn't work can have a large impact on an organization, and it can lead to many problems including:

- Loss of money – this can include losing customers right through to financial penalties for non-compliance to legal requirements

- Loss of time – this can be caused by transactions taking a long time to process but can include staff not being able to work due to a fault or failure

- Damage to business reputation – if an organization is unable to provide service to their customers due to software problems then the customers will lose confidence or faith in this organization (and probably take their business elsewhere)

It is important to test the system to ensure it is error free and that it supports the business that depends on it. The later problems are discovered the more costly they are to fix.

The later problems are discovered the more costly they are to fix.

Some Common Testing Problems

If you make a list of some of the most common test problems, you will realize that in many cases the majority of problems are nontechnical. More often than not, they are consequences of the test process itself, including the overall composition of the test team and whether the company follows well-integrated processes for formal requirements handling and change management. The results indicate the huge discrepancy in the level of importance that different organizations give to testing. Some of these problems are more common to younger organizations; others are pitfalls that anyone can encounter.

The Cost Multiplier

An informal survey of the relative cost of finding defects throughout the software development lifecycle was conducted several years ago. It was found that a problem that goes undetected and unfixed until an application is actually in operation can be 40 – 100 times more expensive to resolve than resolving the problem early in the development cycle.

In traditional testing methodologies, most defects are discovered late in the software development process. Finding and fixing these defects cost much more than if they had been found earlier. Another survey of the relative cost of testing software compared with the overall cost of developing software gives a range of estimates, from 10% in smaller organizations to 70% in some larger and mature organizations.

Lessons Learned/Best Practices

In order to avoid mistakes that may impact the process, cost, and quality of the software testing phase, it is a good idea for testers to get involved with the project as early as possible and consider the following:

- Focus testing on requirements

  Poorly written requirements account for the majority of project failures, so it is important to have the testers involved with the review and test plan creation at the beginning of the project

- Design systems for testability

Systems should be designed and coded with testing in mind. Testers should emphasize the importance of error logs, interfaces and smaller, standalone components which can greatly improve the testability of a system once testers become involved

- Consider usability testing

  One of the most overlooked requirements is "usability". Testers should promote the importance of ease of use and schedule usability tests as early as possible

Traits of a Good UAT Tester

A UAT Tester is one of the most important testing roles since they are validating the system meets the business needs**.** UAT is usually the final activity before the system goes "live" and this role requires multi-faceted skills. These qualities allow the person playing that role to perform this important activity. Without these qualities one may not be able to conduct a proper UAT test. These four core qualities of a UAT Tester are:

- Background: Experience of user operations, Not involved in the overall IT project, Experience in the use of IT facilities, and respected as an independent thinker
- Skill: A good communicator, avoids politics, Expects the system to fail
- Independence: Not involved in user specifications, Has an independent reporting structure, and is a self starter
- Attitude: Lateral thinker, tenacious, analytical

The Role, Activities, and Deliverables of the Business Analyst during UAT

Business Analysts make good UAT testers because they are independent from the developers; therefore arguably more objective. In most cases they also understand the business requirements, and can prepare test scenarios and test data, which are realistic. This allows them to better define the context in which the system will be used and better assess its fit for purpose. Finally Business Analysts have a vested interest in ensuring the system is of high quality so are motivated to perform rigorous testing.

Tasks of User Acceptance Testing

When performing UAT, there are seven (7) basic steps to ensure the system is tested thoroughly and meets the business needs.

1 – Analyze Business Requirements

2 – Identify UAT Scenarios

3 – Define the UAT Test Plan

4 – Create UAT Test Cases

5 – Run the Tests

6 – Record the Results

7 – Confirm Business Objectives are met

Documents Used by the Business Analyst

One of the most important activities performed by the Business Analyst is to identify and develop UAT test scenarios. These scenarios are derived by analyzing the documents that were previously developed during the early phases of the project. These documents include:

- Business Use Case
- Business Process Flows
- Project Charter
- Context Diagram
- Business Requirements Document (BRD)
- System Requirements Specification (SRS)
- Testing Guidelines and Techniques
- Other Vendor's Deliverables

Documents Created by the Business Analyst

Once UAT Test Scenarios are identified, the Business Process Unit will create three deliverables:

- UAT Test Plan
- UAT Test Cases
- After running the tests, a Defect Log captures problems

What is a UAT Test Plan?

The UAT Test Plan documents the strategy that will be used to verify and ensure an application meets its requirements to the business. The UAT Test Plan is a document which outlines the plan for user acceptance testing of the project deliverables. This document is a high level guide, and will refer to test cases that will be developed and used to record the results of user testing.

What are UAT Test Cases?

The User Acceptance Test Cases help the test execution team to test the application thoroughly. This also helps ensure that the UA testing provides sufficient coverage of all the UAT scenarios. The Use Cases created during the Requirements definition phase may be used as inputs for creating test cases.

The User Acceptance Test Case describes in a simple language the precise steps to be taken to test something

What is a UAT Defect Log?

The UAT Defect Log is a document for capturing and reporting defects identified during UAT. Defects are documented so that they can be evaluated and resolved.

Information included in the Defect Log is:

- Severity (e.g., High, Med, Low)
- Status (e.g., Open, Closed, Deferred)
- Date Reported/Fixed
- Problem Description

Summary

Whether your organization designates the functional role associated with testing as a Business Analyst, Tester, or Quality Assurance professional, User Acceptance Testing done well will engage those responsible very early on in the project development cycle. RGF Group Business Analysis curriculum offers many opportunities for learning the skills required of those responsible for UAT.

A little about the Author:

Vincent Bordo is a senior Agile and Business Analyst (BA), Instructor, and Consultant at RG Freeman Group, LLC with over 25 years' experience in the Systems/Business Analysis field. His knowledge, compelling teaching style and use of creative mentoring techniques are his trademark. Vince has presented at numerous conferences, seminars, webcasts, and trade shows, and has developed and rolled-out corporate-wide programs for Business Analysis and Agile/Scrum development for many companies. He was awarded the prestigious *Pinnacle Award* for Customer Satisfaction, an honor that is reserved for the top 2 percent of an elite group of consultants.

Vince uses his years of experience helping companies learn, adopt, and apply state-of-the-art Business Analysis techniques. His skills are unparalleled in Strategy Analysis, Business Process Modeling, Requirements Gathering Techniques, Use Cases, Requirements Management, Data Modeling, and User Acceptance Testing (UAT), Project Management and Agile Project management. He has enabled over 300 companies to effectively apply state-of-the-art Requirements Management techniques to their projects. His compelling teaching style, experienced mentoring ability, and expert knowledge of Business Analysis and Agile has earned him International recognition from both customers and colleagues alike.

Vince Bordo is a Certified Business Analysis Professional (CBAP®), a Certified Virtual Facilitator (CVF), and a Certified Oracle OUM trainer. He holds a BS degree in Computer Science from Rutgers University and a MS degree in Computer Design from Pennsylvania State University

For more information on our **2 day course on User Acceptance Testing for BAs click here:** **2 day UAT for BAs course outline**