A Manager's Guide to User Acceptance Testing

Jan Kusiak, General Manager
IRM Training Pty Ltd ABN 56 007 219 589
Suite 209, 620 St Kilda Rd, Melbourne, Vic. 3004, Australia
03 9533 2300
jan.kusiak@irm.com.au

Introduction

For most businesses and organisations, if IT stops, the business stops. Whenever a company turns on a new production line, opens a new retail store, launches a new product or provides a new service, there is invariably a new or modified IT system behind it. Going live is the culmination of time, effort, resources and finance. A problem-free IT system is the "acid test" of significant, often crucial investment.

Whilst the technical testing of IT systems is a highly professional and exhaustive process, testing of business functionality is an entirely different proposition. Does the system deliver the business functions that are required – does it follow the company's business rules – does it support a government department's obligations - does it cope with exceptions?

The people who have to make these decisions – to accept or reject the new system – are the business users. It is therefore critical to get the business user involved in testing and not rely only on the technicians. In this paper we explore the rationale behind User Acceptance Testing (UAT), why it is so important, and how best to go about it.

Quality is never an accident; it is always the result of intelligent effort.

John Ruskin (1819 - 1900)

Table of Contents

1.0 Introduction – the problems we face

How often do we see project teams making compromises to meet an immoveable deadline? Deadlines can be imposed by business objectives and by legislative and compliance requirements. Whatever the reason, to meet a deadline usually requires a compromise. A brief look at how projects work shows what can suffer.

All projects share three common parameters:

- Time
- Resources (people, funding)
- Deliverables (functionality, quality)

We can't control time so we're left with resources and deliverables. The amount of resources we commit to a project will determine the functionality and quality of deliverables. Functionality by a certain date is what an immovable deadline usually means so we are left with quality. The quality of a deliverable is directly proportional to the thoroughness of testing - but testing takes time.

To illustrate this, chose the answer below that you've heard most frequently:

When do you stop testing?

a) *When all the tests are finished*
b) *When you run out of time*

In an ideal world, all projects would allow adequate time for testing. Project teams would plan exhaustive testing for each piece of system functionality and if they ran out of time then they would drop functionality from a release rather than compromise on quality. In fact this is the case with software developed for products such as mobile phones where the software is embedded on a chip. Most mobile phones don't yet have update capability for their firmware (operating system). If there's a bug you have to replace the chip. Can you imagine the cost of re-calling a few million handsets if there's a bug in the code!!

With business systems, it's virtually impossible to test for every possible eventuality. We must therefore ask ourselves what is the most important functionality that *must* be tested within the available timeframe. The obvious answer is – the business functions that the system will deliver and on which the project justification is based.

User acceptance testing should be performed by business users to prove that a new system delivers what they are paying for. Business users have the knowledge and understanding of business requirements that IT testers do not have. They are uniquely placed to accept or reject the new system – after all they have to live with the consequences.

2.0 User acceptance testing - industry context and trends

Industry today is undergoing a number of major trends that have converged to promote the need for UAT. Let's consider some of these trends and how they can impact upon business.

Globalisation means that more products are becoming commoditised. There are few or no differentiators between products. Any mistake leading to disappointed customers will lead to loss of business. Customers have no loyalty. If they are disappointed they will walk and they will not be back. They take with them today's business and all future business. They also tell their friends.

Outsourcing can be a valid method of reducing costs and increasing productivity. However, companies must be aware that core corporate assets, including customer data, will reside and be protected by overseas companies where data protection laws may not be the same as in Australia. Executives and directors still have the responsibility to prove that they have taken all steps to protect those assets.

Security has become a major concern in recent times with terrorism. This has led to the Patriot Act in the USA and increased security rules everywhere. Accidents with lost and leakage of customer data by partners have occurred and can only re-enforce this trend.

There is now a **critical dependence upon information technology** among companies. Some such as banks cannot survive more than a few days without IT assets in place. This is becoming increasingly complex as supply chains become interdependent upon partners and suppliers.

Formal business processes are being developed and taken up by corporations. Leading the charge are new standards and expertise in formal best-practice risk management. Coupled with this is the adoption of formal business processes in UAT, business analysis and contract management. Executives and directors must take all reasonable precautions (and be able to prove that they did) to manage and mitigate risk to the company resulting from a major change. Changes include the implementation of new software products and enhancements that are critical to the business.

Corporate Governance has arisen from disasters caused by corruption, mis-management or incompetency. Enron in the USA plus OneTel and HIH locally are high profile examples. This has given rise to new corporate standards such as the Sarbanes-Oxley Compliance Standards, the Gramm Leach Bliley Act protecting customer data, the Health Insurance Portability and Accountability Act, and more. There is less impact in Australia as, for example, corporations are only bound by Sarbanes-Oxley if they are owned by or have interests in a US corporation. Nevertheless, Australia has brought in the Corporate Law Economic Reform Program (CLERP 1 – 9) and supported the new globally agreed accounting reporting standards. The trend is set and will only increase as more businesses becomes e-based, globalised and outsourced.

3.0 Case studies – user acceptance testing failures

The following examples provide sobering evidence of lack of planning in the UAT process.

Blue Cross & Blue Shield (Wisconsin, USA) - spent US$200 million on a software development for a new system. Due to inadequate testing, the system sent out duplicate and overpayment cheques to the value of US$60 million. Hundreds of cheques were sent to non-existent addresses, such as the town called "None", due to an operator-input error that was allowed through validation.

The resulting costs were further compounded by 35,000 policyholders switching their business elsewhere.

Bank of New York - In 1985, due to a software error that got through testing, the bank became overdrawn by US$23.6 billion. It was forced to borrow US$24 billion from the Federal Reserve Bank - the largest emergency loan ever made. The overnight interest alone was US$5 million per day.

Clearing Houses Automated Payments System (CHAPS) is responsible for clearing all cheques and automated payments in the UK. Due to a software error that got through testing, the system incorrectly transferred £2 billion to overseas banks. This happened in the space of 30 minutes before the error was detected. The payments were guaranteed and legally irretrievable. The money was eventually returned (as acts of good will) by the overseas banks.

The Jindalee Operational Radar Network (JORN) was a high profile Australian Defence Department project that failed. The review published in May 2000 stated some of the reasons for the failure. There were one million lines of code, developed at a cost of $1.2 billion. Each line of code cost $1,200. Near the end of he project, the software was being compiled and re-distributed every 24 hours. It was expected that User Acceptance Testing would take at least two years.

Failures don't have to be in the multi-million dollar category. A delayed product launch or angry customers is bad enough and can often be the difference between success and failure.

4.0 Business users - the critical factor in UAT

In these examples, we see systems failing at great cost for many different reasons. The systems all went through many types of testing by programmers, analysts and designers. Why focus on UAT? The key is in the word *User*. Quite simply all other forms of testing are carried out by technicians to ensure that the system works technically, according to their understanding of the business requirements.

UAT alone is carried out by the users and business managers to satisfy themselves that the system will meet their business requirements. It is the "sea trials" of the new system. The shipyard can build a new ship for the navy and test every single part but it is not until the navy takes over and tries it out on the open seas that anyone will really know if it will do the job it was built for. Only the navy can tell whether it will do that job. After all their lives depend on it.

Similarly the business users know whether a new system will not only work technically, but also actually allow the company to make money out of it or meet their compliance and legislative obligations. For example, the programmers may rightly say that the new system will allow customer data to be recorded. But will it do it within a two-minute telephone call in a busy call centre? Will the operator be able to record everything in one screen, or will they have to switch between 6 screens to do the job? UAT asks, "We know it will fly, but will it make money - or meet our obligations?"

5.0 User acceptance testing – definition

We can therefore define User Acceptance Testing as:

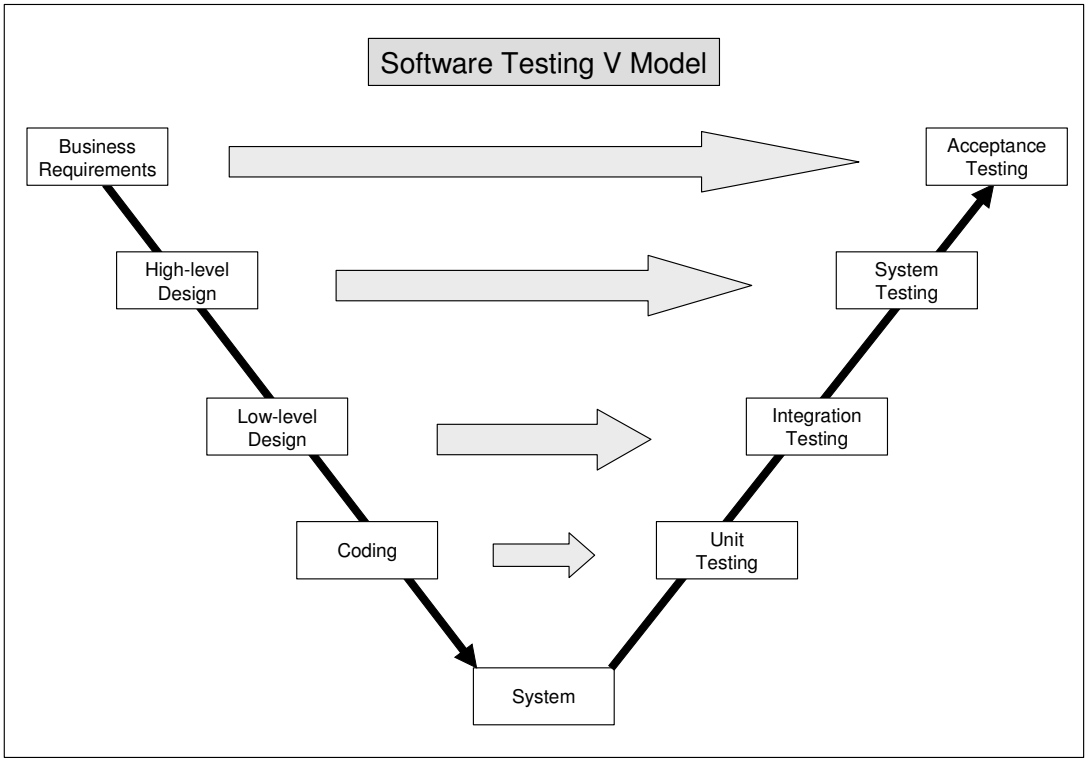
The allocation of resources by the business to determine the fitness for purpose of a new or changed IT system or other asset.

UAT is in essence part of risk management. We cannot allocate time and resources to test everything the system will do. With UAT we must allocate limited resources where they will be most effective. This technique we call Triage. Most commonly the term is applied to medical emergencies where several people are injured and doctors must decide who to treat first. The concept is still valid here and in many management situations. Perhaps prioritisation is a more familiar term.

6.0 Testing – the big picture

One of the most commonly accepted models for software testing – regardless of the development methodology – is the V Model. This maps each development phase of a project to a particular testing phase.

The V Model was originally developed on the premise that there are clear and distinct hand-offs between the different phases of a project. Once unit testing is completed, integration testing follows, then system testing and finally acceptance testing.



Today, this is less clear-cut as unit testing of a software module may involve inputs and outputs from another module. Unit testing therefore cannot happen in isolation and similarly, issues uncovered during integration testing may force a review of unit testing results.

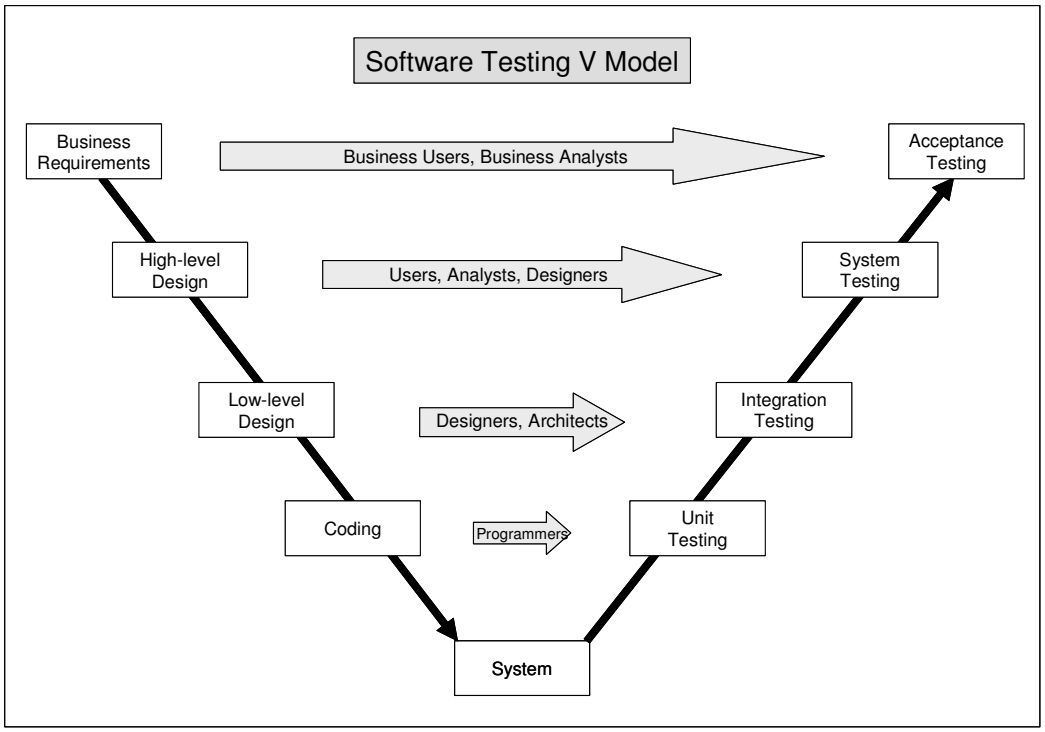
These are issues for software test professionals. We are concerned with testing business requirements - acceptance testing is concerned with *what* the system does. Unit, system and integration testing are about *how* the system does it.

Acceptance testing is the final checkpoint before the system goes live – and before the client pays. Whilst business requirements start the development project, it is acceptance testing that completes it. When specifying requirements our goal should be:

This is what the system must do and this is how we will test it.

7.0 Testing – who does what

Now that we have a clearer picture of the various testing stages, we can consider who does what and the ideal skills needed to tackle UAT.



Unit Testing: Programmers test the individual programs – preferably not ones they have written themselves. The programmer tests that the program can start up and perform its own functions, validate data entered on its screen or create its own data on the database.

Integration Testing: This is the next level of testing that ensures that the individual programs can communicate. It tests that the user can log on and navigate from screen to screen successfully. It does not pay much attention to business functions, mainly the robustness of the system.

System Testing: System Testing proves to the IT project team that the system works according to their understanding of the business requirements. It is a complete test of everything by the project team. At the end of this phase they believe they have done all they can to produce the best system to meet all of the business requirements.

User Acceptance Testing: proves to the users that the system works according to their understanding of their own business requirements. They test the primary business functions they asked for and look at the business results to make sure they are correct. The data input on one screen should be correct when it appears later in a report. The numbers in the report should add up to what the total says. Accounts should balance.

They do not worry about the system crashing; they expect the technicians to have got it to work before presenting it to them (it will get tested for robustness by heavy use in UAT). If it does crash, it is reported to the project team to fix, but is not the focus of their tests. The users also look at business performance. The programmers may say the data can be entered through the screens successfully. The UAT testing asks whether it can be done in one minute while the user is talking to a customer in the call centre.

Note that UAT is the only chance for the users to test the end system to their satisfaction before they commit to using it in production.

8.0 Roles and responsibilities in user acceptance testing

Let's summarise who is responsible for what in the testing process.

- The business unit which owns the system is responsible for testing business functionality
- IT is responsible for supporting UAT with resources, infrastructure and personnel to correct errors
- The testing team is responsible for managing the deployment of resources to provide the best recommendation to management
- Management is responsible for allocating sufficient resources, experienced personnel, infrastructure, time and budget

At the end of the day, management must also make the final decision - to release or delay the system - based upon the testing evidence provided and any identified risks.

Often a system will be released with known risks but an informed decision is always better than an uninformed one.

9.0 User acceptance testing skills

We have already stated that the best person to perform UAT is the business user. However, we should also include business analysts within the UAT team. The analyst will have been involved with the project since conception and will often be the business user's representative throughout the project's life. Whilst business analysts do not normally become involved in the technical design of the system they will often represent or partner the business user at progress meetings and reviews.

Apart from professional credentials and business experience, personal skills are required to make a good tester. These are shown in the table below. Note that as well as good communications skills, clear thinking and a positive attitude, some independence is required.

Not being part of the project team is beneficial (but not always possible). It certainly helps if the tester is not involved in any politics about the system. Essentially testers need to be objective in order to be neutral in the testing.

At least some members of the UAT team should be completely separate from the project until testing. If they have to be involved during the project (prior to UAT) then their role should be passive, such as quality assurance or review. There must be some distance and detachment in their approach if they are to serve the best interests of their organisation.

Background	Experience of user operations	Not involved in the overall project	Experienced in the use of IT facilities	Respected as an independent thinker
Skills	Communicator	Avoids politics	Fence sitter	Expects system to fail
Independence	Not involved in the IT project	Not involved in user specifications	Has an independent reporting structure	Self-starter
Attitude	Lateral thinker	Optimistic and/or pessimistic thinker	Tenacious	Analytical

10.0 Rules of thumb for UAT

Some golden rules to help avoid common mistakes:

- Define testing criteria when defining business requirements
- UAT testers should have their own reporting line - not be part of the project team
- Detachment- during development testers can "look but not touch" (review only)
- Not everyone does testing well
- We are not good at finding errors in our own work
- Teamwork is essential
- It's almost never right first time
- A successful test is one that finds errors

Finally, keep a positive attitude; testing is not about assigning blame. Quality standards (process and procedures) should ensure that problems are identified and avoided at source. Testing is the safety check that should catch any errors that slip through.

I have not failed. I've just found 10,000 ways that won't work.

Thomas Edison (1847-1931)

11.0 Summary – making UAT a success

Remember that user acceptance testing is a business, not a technical, activity. The following points will help in keeping this prominent:

- UAT is partly a risk management process, a balance of risk against available resources
- UAT is the business users checkpoint, it tests the business outcomes
- Resource planning and prioritisation are required, not everything can be tested
- Testers need experience and knowledge of business operations and business rules
- UAT needs people working together – business users, IT, suppliers and even customers

To re-emphasise the final point, the system supplier (in-house or external company) must work with the business users to develop a testing strategy, valid acceptance criteria of the business functionality and a test plan that maximises the potential for a successful system rollout. The business depends on it.

Send feedback and comments to: training@irm.com.au