

te testing experience

The Magazine for Professional Testers

Test Techniques in practice -

Do they help?

Why do we often test without them?



Software Testing – the future?

by Geoff Thompson

When I was set the challenge of writing an article on the future of software testing, having got over the initial creative panic, I was wondering how to start. Then I found the following headlines in two well known online computer journals:-

'BA: software failures key in T5 fiasco'

Computing 8 May 2008

'Update: lack of software testing to blame for Terminal 5 fiasco, BA executive tells MPs'

Computer Weekly 9 May 2008

In summary both articles made the same point. During questioning by MP's Willie Walsh (BA Chief Executive) reported that construction work had not completed in time squeezing the window for any real testing. The management met, identified and agreed the risks and issues this creates and decided that the risk was acceptable to enable a go live as planned.

In hindsight Walsh said "If I were to pick one issue I would have done differently, it is that, having recognised the importance of testing and having designed six months of testing, we subsequently compromised on that."

Computer Weekly 9 May 2008

Why, did it interest me? Quite simply the management of the project did exactly the right thing; project management and risk management were used to assess whether slippage in earlier parts of the lifecycle meant that reducing test activity and timescales would have an impact on the quality of the finished article. Obviously the risk was perceived to be low as a decision was taken to reduce testing and go live on time (a decision now regretted by Walsh).

What I also found interesting was that one of the papers saw software testing as the issue.

Initially I thought here we go again, testing put the bugs in etc. but then I understood. I wonder if it was more that there was a lack of the right information from the test organisation that would have enabled the right decision to be made by the BA and BAA management. If the test organisation had a grip on the real issues that mattered to the management team and were able to communicate the real risks maybe a different decision would have been taken, or at least the decision that was taken would have been made based upon the right information, with complete awareness of the impact.

To me, testing's role is the provision of data to enable go live decisions to be made (I have heard this termed as Trusted Advisor and Project Intelligence). Yes we might be involved in the decision but ultimately the decision belongs to the Sponsor and his management team, but the data used in the decision will predominantly be derived from test activity and must be sufficient to enable the right decisions to be made.

Given this is a major element of our role; I suggest that we need to get this right or the future of software testing will be very dim. We are in the best position, or at least we should be, in a project to gather real data, and to establish what the right data may be, to enable the accurate reporting of project progress information. Far too often today a test team arrives and sits in its cocoon only appearing at the end of the test cycle to announce they haven't finished but can't give anything helpful to say other than perhaps statements like:

"You can't go live because..."

"We have 150 tests still to run"
"We are still finding errors"

"I wouldn't go live yet"— yes I heard that said by one Test Manager one day – and this was all he said

None of these provide even the slightest hint of what the impact of going live is! How critical are the 150 tests left outstanding? What's the error finding profile? Why is testing still finding errors now? These are just some of the questions that these statements generate but can't answer.

So, if we get the information wrong, wrong decisions will be made. There was however one very good example of a situation where the right information was provided in the wrong way, and therefore ignored. The 1986 Challenger disaster was predicted – if you knew that the Space Shuttle would definitely explode killing all of the astronauts, would you have given the go ahead to launch. NASA did. Several engineers had voiced concerns about the effect of the temperature on the resilience of the rubber O-rings that sealed the joints of the Solid Rocket Boosters (the tubes attached to the side of the Shuttle – that are full of fuel at take off). They argued that if the O-rings were colder than 53 °F (12 °C), there was no guarantee they would seal properly. This was an important consideration, since the O-rings had been designated as a "Criticality 1" component—meaning that there was no backup for them and their failure would destroy Challenger and its crew. They also argued that the low overnight temperatures would almost certainly result in Solid Rocket Boosters temperatures below their redline of 40 °F (4 °C). However, they were overruled, the launch happened, the O-rings failed leaking rocket fuel, and the rest is history.

Interestingly the right information was identified but if I am very critical, using words

like ‘no guarantee’ instead of ‘an absolute guarantee’ when the management board were under significant pressure from the US Presidency to launch, didn’t really emphasize the facts strongly enough. In the same situation, with the same information, I might well have also decided, like NASA did, to launch. Now I know it’s a word thing, but in reality if we can’t express ourselves and ensure that our message is understood all the great work we have done to improve the processes we use and align vocabulary will be wasted and we will never grow as an industry and will always be seen as the also ran profession! It’s all about communication.

Sometimes, alright I concede, most of the time the information we need to provide is not what the Programme wants to hear, but it’s our job to ensure the right information gets through to the right person and that they are able to understand and make informed decisions. We need to be prepared to be the bearers of good and bad news.

If testing can’t provide the right information it actually adds little value, and that I suggest maybe the situation we find ourselves in a lot of organisations today, we all work very hard at our profession but no one outside of it seems to understand what we do and why.

Whilst working on one client site I remember interviewing a Test Manager who was most upset that the Programme team had asked him to stop issuing his daily report, for what he described as stupid reasons. When I met and interviewed members of his programme team they said that the report was stopped because from day one the Test Manager had had the project flagged as red, what was requested was that the report included relevant data (such as where the issues lay, what element of the software was not working etc.) or was stopped. The Test Manager saw no reason to include the extra data so stopped issuing it. Even today the Test Manager cannot understand what he did wrong and blames the programme team for the delays in delivery because they did not listen to him!

If Software Testing is ever to be seen as a profession and be respected in the future then we should focus on the information we need to enable the Programme to make the right decisions.

Now I am not suggesting that I know the answer. Each person will have differing requirements, and most importantly their requirements may be very different to what they request individually when asked as part of a team (I call this the pack syndrome – when a group influence individual perspectives and results. If you get separate agreement to an action from each individual in a group of people, when you put them together the actions may and often do change), but it is our job to identify what they are and focus our approach to testing in order to deliver these. We need to be conscious of the pack syndrome.

The data requirement is never static, so we also need to be conscious of the changing needs for information as a project progresses.

To enable the future of software testing we have to influence people’s approaches to Test Management and the provision of information? Is it a qualification thing, or maybe standards, there are test models out there and what role do tools play in this arena? The following sections look at each of these individually and their relevancy to the issue.

Software Testing Qualifications

You may or may not know that I have been involved in Software Testing Qualifications for a few years now! Firstly working with ISEB (Information Systems Examinations Board) and lately internationally with the ISTQB (International Software Testing Examinations Board). So I guess I am a little bias, but I believe there is value in them. However, there does seem to be some polarisation in the industry as to the value of these types of qualification, some calling for them to be banned, others suggesting that having got them you are able to sell yourself as a competent tester.

The reality is that a good tester or test manager needs many components of skills and knowledge combined to make them successful. In my view the qualifications account for about 10% to 15% of the knowledge required for the practical application in the workplace.

At ISTQB Foundation level there is a short training course (3 days) followed by an exam. However the quick learners amongst us could very quickly understand what is required and pass these exams, maybe even without ever testing anything, there is no minimum experience requirement to take the Foundation.

Intriguingly ISEB have recently revised their single Practitioner Exam, into three, the Intermediate (a mandatory common module which contains their view of the generic detail that is required for the advanced level), Test Management and Test Analyst, but only prescribe a 3 day training course followed by an exam. There is a minimum entry requirement, but they do not include practical test experience, more a passing of time between taking the Foundation exam.

ISTQB, on the other hand, have just launched their Advanced level role-based qualifications, and like ISEB there are three of them, Test Manager, Test Analysts and Technical Test Analysts. However unlike ISEB there is no common module, as these are role based (for example the role of a Test Manager in a review is very different to that of a test analyst) with a minimum training period of 5 days each followed by an exam, with basically the same entry criteria as ISEB.

So I believe qualifications have a very important part to play in the education and verification of software testers and test managers. But

the reality is that the exam candidates leave with an understanding of the basic vocabulary and approach to testing, but in most cases no real practical capability, and certainly no understanding of how to communicate outside of the test team.

Standards

There are a few out there, for example BS7925 – 1&2 (the British standard for Component Testing and its associated glossary), IEEE829 (the Test Documentation standard) etc. etc.

In their favour Standards are developed by a very wide working group (these can be as many as 1000 contributors) who build and review the content. In most cases involvement comes from right around the globe. However, by necessity they are generic and some concessions have to be made to gain global agreement (in fact I understand that the latest version of IEEE829 couldn’t be released as planned as the detail was not agreed) and provide one solution for everyone, so they are useful but they can’t provide the whole answer.

Standards don’t, in any example I have seen, indicate how to provide and what information is needed to communicate effectively with the programme, although they can help. For example in IEEE829, when it discusses entry and exit criteria and closure reports looking at planned activity against actual activity and lessons learned, although it mentions these key elements of a test project it doesn’t provide any help in pinpointing the right detail.

They are however excellent process models, that provide guidance on how to establish your test project.

Test Models

There are many models available for testers to review against. Most consultancies will have a Test Maturity Model of their own, most sadly leading to creating a sales opportunity, and not actually there to help the organisation that has been reviewed against the model to improve what they do. The nearest commercial model that really does set out to help companies is the TPI® model provided by Sogeti. However this is now a little out of date; it is based upon their in-house test method T-Map which has been updated and those updates have not been reflected within TPI®.

Probably the best model available today to help ensure good processes in use across test projects is the new TMMi Model. This model has been developed by the TMMi Foundation (www.tmmifoundation.org), an international organisation whose aim is to establish a non commercial test process assessment model for the industry. The TMMi model takes its input from many places including the original Bernstein TMM, and others including the Gelperin and Hetzel’s Evolution of Testing Model [Gelperin and Hetzel], which describes the evolution of the testing process over a 40-year period, Beizer’s testing model, which describes the evolution of the individual tester’s

thinking [Beizer], research on the TMM carried out in the EU-funded MB-TMM project, and international testing standards, e.g. IEEE 829 Standard for Software Test Documentation [IEEE 829]. The testing terminology used in the TMMi is derived from the ISTQB Standard Glossary of terms used in Software Testing [ISTQB].

The model is structured much like CMMi with 5 levels of maturity. At level 2 of the model projects begin to be assessed on their capability to measure progress through testing. However like the CMMi model it is not prescriptive as to what is measured just that measurement occurs, although help is provided regarding the minimum is expected.

Tools

Now to some these are the panacea, I prefer to look at tools as the slaves of the process. When it comes to information provision there does seem to have been thought put in most examples I have seen, to the data that the tool provides. The Dashboard seems to be the latest fashion accessory for any company wishing to set its stall in the test management arena. HP has one in Changepoint, Compuware in ODM etc, etc. In principle these provide data derived from actions within the system.

Tools can be excellent providers of data, but that data is only as good as the way it is collected. A common problem, for example, with Test Management tools is that the tool is implemented straight out of the box, with no configuration; testers then load test cases into the tool with no views on how progress will be tracked, so they load the minimum data to allow them to do their job. Then they try to track progress, but realise that the data needed to do this hasn't been loaded, so they can track certain elements, so unfortunately in my experience the reporting in these situations is very poor.

So with some real thought up front as to what information/data is required the tools can be configured to ensure that data is a by product of the process (e.g. does not become time consuming to collect and report).

Each approach listed above contributes in a small but very useful way in helping the Test Manager recognise the data that is required, and to some extent the way that data should be communicated, however this doesn't extend across the different roles that exist within a project. It doesn't prescribe what it is a development manager needs to know or the Business Sponsor.

So having identified the data, the key is how it is presented back as information in the right format at the right time.

This particular issue does seem to reside in the Test Managers camp, so I would like to focus there for now.

I believe that the future of testing depends on

the following:-

- A. Accurate information provision, in the language of the recipient
- B. Building quality in, rather than testing it out, prevention rather than detection
- C. Ensuring that quality is considered and not seen as a burden, alongside time and cost, which seem to have become the new buzzwords

As I have said already the great thing about Qualifications is that they have established some common terminology amongst testers, but that terminology is still not translatable across the lifecycle. So we have four languages as a minimum being spoken in projects today, we have the Business (client), the designers, the developers and the testers, and in an ITIL (IT Infrastructure Library) organisation is possibly a fifth.

We have projects that sit together but don't talk to each other. We also have so called Agile projects starting to show how greater collaboration brings big benefits, is this the way forward?

I think the way forward for information provision is actually to learn something from the Agile world and start to talk to each other, and not to work in silos. As I have already explained, everyone's requirement for information is different and so the closer projects work together and people mix the more we can understand what our requirements are.

In a world of increased outsourcing or offshoring this becomes increasingly harder to achieve. This is especially relevant when the test resources, or development resources, work across the other side of the world in a different time zone. It is as important, if not more important, that in these situations proper and regular communications are established. It's not easy but if success is expected then effort needs to be put in to ensure that each party is aware of their goals and provides the right information to enable measurement of progress to be tracked and understood.

Lots of organisations seem to think that by putting people in buildings in different countries and arming them just with a process they will get great results. This ignores the need to work as a team and communicate, and so in most instances is a strategy doomed to failure.

I think it worthy of a stop here to review another element we need to consider. We (and in this instance I am not just referring to test resources, I refer to anyone involved in a software development project) need to accept that Software lies at the heart of almost every business process. Everything from how an organisation communicates internally and externally, to the way it delivers products and services, to how it manages administrative processes. Software can liberate or debilitate an organisation's efficiency, effectiveness, risk exposure, risk exposure, and ultimately, its profitability.

So it's not something a Test Manager can do alone, we have to work together to ensure that in the increasingly complex world of software development we understand what it is we all need, and ensure we deliver it.

If the test manager for Terminal 5 had provided the following information to Willie Walsh at the point that the decision to reduce testing occurred I wonder if the result may have been different.

Test Manager –“Mr Walsh we have reviewed progress and remaining activity and can report that should you reduce testing in the way suggested the following will happen:

In the first 5 days of operation:

1. Over 23,000 bags will get lost, leaving you with a bill well in excess of £300,000 to reconnect them with passengers
2. 500 flights will definitely be cancelled
3. Staff will not be able to park due to issues with the security system, and so will not arrive on site on time
4. 28 lifts will not be operating
5. You will definitely be pulled up in front of the UK Parliament to explain yourself

I recommend we delay launch for a minimum of a week so we can resolve these issues.”

Would he have agreed to the same launch date? That I cannot say, but at least if he had, he would have understood the issues he was to face with the press and public.

But you may say hindsight is a great thing. If we could all see into the future we could perhaps all determine these types of issues easily, but how do we do that prior to delivery?

Project delivery is not difficult, it is simply the bringing together of professionals in their field to deliver to a single goal. If that team works together then, for example, how much can a developer learn from a business analyst if they talk about how the system should actually work, or where it always doesn't!

A project manager who just cares about meeting delivery dates regardless of setting, and meeting success criteria is a fool in my book, and one that shouldn't be working in our industry. Success criteria are the goals that are the starting point for information; and may be different for each different level in the project, for example:

1. Success for the Sponsor may be a system that saves or makes them £3bn a year. In that instance the information provided needs to include data to reflect achievement when it goes live
2. Success for a Test Manager may be that there is an agreed risk based approach to testing established at the outset of the project, and all high priority test cases are agreed up front and are completed during

test execution, but let's think about that. So to be successful a Test Manager needs to have a prioritised list of test cases, which they ensure are completed during test execution.

This is an altogether too familiar success criterion today but maybe one that doesn't help point B above (building quality in, not testing it out), maybe a better success criteria could be:

- a. Test activity defined that reducing the amount of test execution to one week (a tall order but if the team works together and identifies up front where things may go wrong and stop them occurring. Using an analogy, if you knew early enough that your wall was about to fall down, would you write a test and then go away until it fell? Then check it had, raise a bug report and pay a lot of money to have it repaired? Or would you do something as soon as possible to stop it falling and keep the cost, time and quality at the right levels? Translating that into a project, if we know the objectives of a project manager include time, cost and quality we need to find ways of preventing rather than detecting issues as this is the most economical approach to delivery, and to meet all three objectives).

Test managers and project managers must also reflect progress towards achieving these criteria. But we shouldn't forget that throughout the lifecycle of a project these will change as each new issue and risk is resolved, and so by necessity, must the information provision.

Teamwork is absolutely key to achieving project delivery in the future, and ensuring software testing continues to deliver value. Sadly this is something that gets very little daylight in IT. We ensure all of our people are technically competent but we don't really consider that they need to be taught how to manage people and how to build working teams. Good developers are promoted and asked to manage people, and it's the same in testing. How many testers ever get sent on a project management course to learn the basic project management processes? Very few, in fact in an organisation I worked in it was considered a course that was too advanced for test resources to attend! However developers and designers aspiring to become Project Managers were allowed to attend.

I think a software delivery project is much like an orchestra, if each part works together in harmony it can be beautiful, but if only one element works on their own the result will be disastrous. We need to work together so that the software delivery orchestra works, and delivers beautiful results.

So now let's go back to where I started. Good information provision comes from being part of a team who communicate regularly and fully understand the progress through the different lifecycle stages, all the way through to the live environment.

At this point I can't help but remember the quote – “You cannot manage what you cannot (or do not) measure”. (anon). So we must know:

- What do we need to measure and does the information and importance of the information change during the lifecycle of the project
- We must still capture the number of test cases, tests run, tests passed/failed, but need to identify how to translate this information for the different audiences we have
- We need to capture 'quality requirements' and therefore identify how to measure meeting them. They must be clearly and concisely defined, easily understood, not ambiguous and easily demonstrated
- Testers must encourage the breakdown of the silos. Why not work within the development team and help them test?

No one can define today what the future of testing is but by learning to communicate more effectively and by becoming an integrated element of the project, I believe the future of testing is assured.



Biography

Geoff is Consultancy Director of Experimentus (www.experimentus.com). He has been actively involved in testing for over 20 years covering Test Analysis, Test Management, Test Programme Management and Test Strategy Development.

Geoff is a trained Test Maturity Model (TMM) assessor and can perform Test Maturity assessments. Obtaining the accolade as Test Manager of the Year 2004, Geoff was also one of only three testers in Europe nominated for The European Testing Excellence Award in 2003.

Geoff is a recognised thought leader in his field. He is a founder member of the ISEB Software Testing Board, International Software Testing Qualifications Board (ISTQB) and the TMMi Foundation, of which he is currently Treasurer. Geoff was directly involved in the creation of the original ISEB Foundation syllabus in Software Testing in 1999, managed the delivery of the ISEB Practitioner syllabus in 2002 and is currently the UK representative to the ISTQB - formulating and managing the new Foundation, Advanced and Expert level syllabi.

Geoff is also Chairman of The UK Testing Board and Vice Chairman of the BCS SIGiST (Specialist Group in Software Testing).

Geoff's expertise is widely documented through articles and publications. He co-authored what is currently classified as 'the best selling testing book on Amazon.co.uk', ISEB – A Software Testing Foundation. Geoff is also a frequent speaker at major industry events nationally and internationally.