



Test Planning and the Future

A new look at an old friend

Our Plan for today:

Introduction

- Quick bit about me

Test Planning Today

- The general, one size fits all test plan and what issues we see today.
- What motivates us to do it this way anyway?

Re-Inventing the Wheel

- Some ideas on what artefacts you could create instead
- Building artefacts that have lasting value to your development/testing teams

Implementation

- What are some of the things you should consider to implement a different approach

Questions and Answers

- Have at the madman who dares assail our sacred document!

About your speaker

- Been a tester for the past 10+ years
- Currently the Director of Quality Assurance at Navantis
- Have tested a variety of software/hardware platforms
- Have spoken at QAI in Toronto/Chicago
- Also founder of the Visual Studio Tester User Group for Toronto.
- Openly admit that I still write test plans!

How is test planning “typically” done in organizations today



TEST PLANNING TODAY

Creating a Test Plan Today

- To be successful, we have to rely on our team members (PM, BA, Dev)
 - Bad Requirements or Architecture likely = bad test plan (but that does not have to be the case)
- Our typical required output: A monolithic word document
 - Tends to include environment, tool requirements, resourcing, scope, risks, entry/exit criteria and so on
 - Often based on existing standards such as IEEE, ISO and so forth (or copied from the template you used at your last job!)

What makes a plan good?

- A good test plan/strategy is no different than any other project plan:
 - It needs to clearly define the testing mission, what it will measure and how it will be measured
 - It needs to provide clear direction that can be distilled into logical next steps in any testing activity:
 - Test case writing
 - Environment Setup
 - Entrance/Exit Criteria
 - Automation/Performance/Security...And so on.

Some issues that often arise

- Plan tends to be sacrificed at the first sign of danger!
 - When things get busy the test plan is never to be seen or heard from again
 - What is the real long term value of such a document if it is the first thing tossed off the bus?
- Is considered a “waste of time” by some testers
 - Do they have a point or not?
- Often can lack important details due to it's attempt to be a one size fits all document
 - Very heavily dependant on the ability of the author to speak to the various audiences who ***Should*** be reading it.

So what motivates us to do it, really?

- Because it's important! Silly!
 - Completely agree – it is important to plan a test effort but is it important to everyone?
- Because it is a deliverable as part of our project
- It is something we need for client sign off
 - Generally our clients know very little about testing and fixate on aspects document they do not understand.
 - What do you mean by “Risks”
 - How come you are not testing every OS/Browser combination
 - What are these tools your talking about .. Do I have to pay for them?
 - Why are declaring that this functionality is less critical than another.. Everything is important!!
 - What do you mean there will be a defect/bug in my software?
- To cover myself from the inevitable “why did you not test x or y” questions
- To advocate for our needs, to make a solid argument for a given test approach
- Because that no-good, do nothing QA manager told me too!
 - If he would just let me spend my time writing test cases everything would be so much better!

In Summary, today we..

- Write test plans because they are important (and because we have too!)
- Generally create a large, all in one document that has many intended readers
 - But few read or pay attention to our hard work
- Have a hard time keeping it in sync with project changes
 - Particularly in rapid SDLC's (Agile, Scrum, XP)

How many ways can you slice a Test Plan in 10 slides or less?



RE-INVENTING THE WHEEL

So okay smart guy.. I'm listening.

- Okay so lets think about this for a second:
 - We have an artefact that has to:
 - Speak to many potential audiences (even if that is not the intention)
 - Needs to live and breathe with the project
 - Has many key sections covering everything from scope, tools, environments, acceptance criteria into and out of test, and what day the QA manager will go crazy and so forth.
- So lets look at what can happen when we gravitate to appeasing a particular audience

Test Planning – Forces that can effect the Quality of the Plan

The reality is a good test plan advocates for all these interests, but maybe this is actually 3 or more things we are creating here?

When Test Plans are written for Clients/End User Audience:

- Test Plan becomes a mechanism to “make client happy”
- Test Plan reads more like a requirements specification than a test plan.
- Test Plans are missing key technical planning points that are important to the testing team since they will “confuse the client”
- Can at times seem more like a user guide than a test plan (e.g. How to use our bug tracking system)

Business/End Users

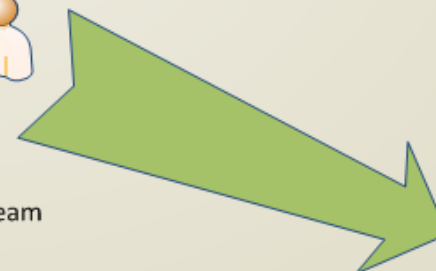
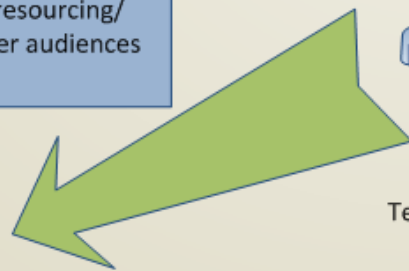
Risks of writing a plan with too much Organization focus:

- Plan is merely treated as a project deliverable, with little meaning beyond that.
- Plans can become very generic with little thought given to the best approach for a given project.
- Testing strategy focuses more on resourcing/timeline than important details other audiences might need



Testing Team

Organizational/Project focus



Testing/Technical Focus

When the plan is very technical:

- Test Plans written this way often get mired in exhaustive details that are hard to maintain.
- Test Plan becomes more than a plan, often explicitly writing test cases in the document
- Can fail to stress important details that the end users or clients might care about, often lost in automation strategies and tools
- If it is a client deliverable – they often do not get it and will need a great deal of hand holding

What Deliverables we might create instead:

The reality is a good test plan advocates for all these interests, but maybe this is actually 3 or more things we are creating here?

1. Since part of test plan focuses on milestones – why not work with PM on project plan?
2. Use to map out discrete testing activities into tasks (see below)

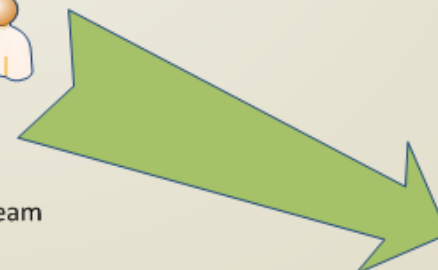
1. How about a light scope document for the client about what we will test?
2. What about a standard user guide for our tools and systems?

Business/End Users



Testing Team

Organizational/Project focus



Testing/Technical Focus

1. Create distinct Tasks for your test team members in your organizations system
2. Create Environment Diagrams and store them on central project location

Sure make more work for me!

- Okay – so breaking this into 100 separate pieces looks like more work
- Actually it should not be, in fact there are several opportunities to streamline and enrich the process, providing longer term entities which can be of longer term value to everyone.

A quick example of how this works:

- Imagine your typical Test Plan Table of contents..
- As you can see you have a number of entities in the document that may be able to be logically split into something other than a section in your document.
- For example, your scope section might logically become a client facing high level plan on its own.

What about all the other sections?

- Depending on the system you use, you could easily morph these into your project plans and/or real tasks for your team to bite into.
- Environmental diagrams, exit/entry criteria's can all be documented in a different way.
- In the end you should end up with more entities, but easier to find and maintain what you need to as the project moves forward.
- Lets take a real look at what that might look like

Table of Contents

1	Document History	3
2	Introduction	3
2.1	Purpose	3
2.2	Intended Audience.....	3
3	Bounds – What we will and will not be doing	3
3.1	Scope – What we will be doing.	3
3.2	Unit Testing – “You mean you actually unit test?” Yes and enforce it too!	4
3.3	Workflow Test/Functional Tests: Our bread and butter.....	4
3.3.1	Browser Compatibility Testing – Google... I hate you.....	4
3.3.2	Solution Deployment – You mean something could go wrong with deployment?.....	4
3.3.3	QA Testing Priority Matrix – All function points in the application are created equal, please disregard this section – your customer.....	4
3.4	Acceptable Testing Conditions (UAT) – Clients *LOVE* this section. What do you mean my software might have defects?.....	6
4	Risks – The Sharks hovering about that hate testers	6
4.1	Quality Risks	6
4.2	Schedule Risks	6
4.3	QA Manageris Nuts	6
5	Schedules and Milestones – The mini project plan we create and commitments we are making.	6
6	Test Configurations and Environments – And the clients begin to glaze over here...	6
6.1	Software Requirements – yes QA actually needs pliers, wrenches, callipers and hammers to do our job. Okay we prefer the hammers.	7
6.2	Hardware Requirements – Please host on PC, not MacK-thx. *Note no Macs were harmed in the making of this test plan.....	8
6.3	Project X Test Environment Diagram – IT departments cannot read so draw pretty pictures instead (nor can QA managers).....	8
6.4	Resource Requirements – As for 1000 hours and get told you have 10.....	9
7	Test Execution – More detailed task breakdown of what we testers really need to do	9
7.1	Pre-Deployment Testing Schedule (tentative) (and therefore will likely change and nullify everything in this table.....	10
7.2	Automation Strategy – What do you mean you need “time” to automate?.....	10
8	Entry/Exit Criteria - We do not accept stinky piles of goo into QA, and it's a stretch but we don't think the client will like that either.	10
9	Referenced Documentation – Testers read too	10
9.1	Traceability Matrix – Yes I really actually based this plan on something other than my 4 year olds imagination.	10
10	Definitions – please, please explain those damn acronyms. I am not in the army	12

Sections 1-3 could easily become a lighter, testing scope document alone

Section 4 – Risks could be added to existing risk registers for all to see

Pictures say 1000 words: Draw and share your vision of the test environment.

Test Execution can be part of your project planning / task creation

Entry/Exit Criteria can be created and flagged for each build or made global



How you could potentially implement this approach



IMPLEMENTATION

Implementation Steps – Potential Roadblocks

- Testing group has to believe and envision how this could work with your existing tools
- Need to sell the powers at be on a different composition to your test plans – this may include customers, internal executives and so on.
- If your plans are not being read today, making changes may not be too difficult.

Implementation Continued

- Pick a trial project to try this out on
- Iterate and refine your approach over time
- Never lose sight of the goal: to make the results of test planning important and useful for the duration of the project
 - Make sure your artefacts are not onerous to maintain, that they are being read/used.
 - Make sure they are useful to their intended audiences
 - Remove anything that is not providing immediate and lasting value to the process of testing.

What if we need to keep the beast?

- If a formal test plan is still required for any reason, consider how this could be assembled quickly
 - If we can automate some of the content population (based on the ideas presented today) perhaps you can have the best of both worlds.
 - This could also make that test plan a true living document!

Questions and Answers

If there are any thoughts/ideas you have around test planning, I would love to hear it.

I can be contacted at:

raifm@navantis.com

Director Quality Assurance

Navantis, Inc.