

# 21st century schizoid plans

Technical consultant **Gordon Alexander** of Compuware believes resistance to documentation and procedural standards is rooted in lack of sufficient training, resources and time to implement them properly

*I talk to a lot of people about testing.* When you talk testing you talk documents; lots of them. There are different thoughts on handling this.

Sometimes I'm told that things aren't being done correctly, a feeling that procedures and documentation are a bit flimsy, not as formal or organised as they should be. These are the guilty, furtive testers muddling through in fear that they are about to be found out.

I hear the contrary view too. Heartfelt complaints about documentation and procedural standards foisted upon an already overworked organisation. Restrictive and pedantic without adding anything of value to the development process. These are the wild free spirits held back by the chains of bureaucracy.

Of course, both these views are perfectly reasonable but the bizarre thing about this is that very often *it is the same people who hold both these views simultaneously*. Analysing why these multiple personalities exist within the testing practitioner community is actually quite illuminating.

## What exactly is the testing problem?

Without documentation and standards you discover a whole set of issues:

- Everyone has to re-invent the wheel;
- People may or may not know what to do or how to do it;
- You duplicate work that other people have done;
- You repeat work that you have done;
- You don't know what you have done. Eg, you can't tell people what you have and haven't tested. "Is the warp drive going to work up to warp factor 10?" – "Erm, don't know, let me ask Scotty if he checked that."

Unless you can consume a lot of their time it is difficult to capture input from other groups to assist and inspire your testing.

The purpose of CMMI, SPICE, TMM, TPI, Six Sigma and all the rest of the many and varied methods that exist today, is to deal with these issues (amongst other things). However, introducing documentation and standards can initiate a whole set of new problems:

- You may not know where to start to create the documents and standards from scratch;
- You don't have time to set it up during this release "We'll do it next release when the fires are out";
- It would take more time than the whole project to design and develop a decent set of procedures;
- Using standard methods you find that many of the steps in the procedures and much of the documentation does not add anything of value to the project;
- It takes an incredibly long time to produce all the documentation even though it is of some value;
- You spend all your time doing documentation instead of testing.

## So, what is the answer?

What is the best way of organising your testing? I've seen a lot of good testing being carried out by simply using common sense and a high degree of individual skill. This can work well, but it's not optimal.

Sometimes I've seen people attempt to move towards a more formal approach but have encountered difficulties in doing that. Implementation often founders because efforts are made to implant an entire process onto overworked people who aren't adequately trained or resourced to implement it. This is why you can sometimes find our testers simultaneously expressing seemingly contradictory views.

In my opinion, the best way to test is to implement a tailored and resourced standard method using well-trained staff given time to get through the learning curve.

So what do we need? Documents.

Let's have a quick look at what sort of documents we are talking about. Broadly speaking, there are three main categories of documentation.

- 1 Documents to manage the process of testing: *test plans*
- 2 Documents that describe the tests you wish to run: *test assets*
- 3 Documents to record testing results and faults: *test logs*

## Test plans

Here I mean things like test mission, test policy, test strategy, test plans, test schedule etc. A fundamental tenet of testing is that possible activities can consume an infinite amount of resources. You use documentation to manage this effectively. You also use your test planning documentation to write down what you mean by testing, thereby limiting the scope, and to say how and when you are going to do it. The reason you write it down is so that you can get it clear in your own head and get everyone else in the organisation to agree it. This is particularly important when looking at risk.

It is interesting to examine all the different documents that exist in this category. Sometimes it seems that they are duplication. They all purport to explain what, when and how you are going to do your testing.

Most times you need to design a hierarchy of documentation, which is why you need more than one type of document in your 'test plan' set.

You have two different kinds of information that you need to write down.

Firstly, there is information about testing which is applicable to your whole organisation. This is general information. This includes things like the purpose and definition of testing, the methods that are available for use, the measures that will be made and the tools that will be used.

# Are your applications leaking money?



## 64%

of IT Executives say Yes...  
*having experienced  
material revenue loss  
as a direct result of  
application failure\**

Compuware's ASQ solution can prevent application failure by building quality in from the start. To learn more about how we can help save you time and money simply visit

[www.compuware.co.uk/products/qa](http://www.compuware.co.uk/products/qa) and request your FREE Patricia Seybold Group White Paper - 'Your Applications are Leaking Money'.

Alternatively you can call us on 01753 444 444

In addition you can view our latest webinar 'Quality First, Strengthening the Quality Process for Successful Application Deployment' by visiting [www.compuware.co.uk/webinars/asq2](http://www.compuware.co.uk/webinars/asq2)

COMPUWARE®

[www.compuware.co.uk](http://www.compuware.co.uk)



These can be very different for different organisations. So for one company the goal of testing will be to find bugs, for another to improve quality. The communication of this information across your organisation can have a profound effect on the testing activities, for example the goal of “proving the software works” is vastly different from “reduce the number of failures that occur in released software”.

The other type of information that these documents contain is project-specific information such as who will do the testing, when it will be finished etc. It can also contain information about variations from the standard rules of the organisation. You should keep clear in your mind these two different purposes when writing your documents.

So how do you decide what documents to have in your hierarchy and what to put in them? For the general (non-project-specific) information each higher level document contains a summary and more a strategic view of the information. So for example, a test strategy document may say “we will proactively investigate the use of test tools provided there is a demonstrable return on investment in six months,” whereas a test plan may describe specific tools to be used.

The decisions you have to make are whether to have a particular document in your hierarchy set and what information to put there. Use the standards to get a view of everything that anyone could possibly want in documentation. Then the key is to be really honest about the

audience and the purpose of the document.

So for example, if you want to provide guidelines to your team of four testers then you don’t need a test strategy, you need something more specific, “use state transition testing, equivalence partitioning in your testing” etc. If, ultimately, you can’t influence other testers in your organisation, then there is little point in spending your time simplifying, generalising and explaining what you do in a document that no-one except you and your team will read.

If however, you want to ensure a certain level of quality across all key technologies in your company, then get your best people to write a strategy in terms that you, the board and all stakeholders can understand. This strategy should be both a condensation of your current best practices and your best vision of the way forward.

You can give access to all levels via a simple but high-level interface which details the test status – with red lights representing significant issue areas, yellow lights representing areas of caution and green being fine.

Similarly, if you look at project information each level of the hierarchy contains more detailed information about the project as it is developed and more details are known or required.

When defining your test planning documentation keep these principles in mind and you won’t go far wrong.

## Test assets

Generally tests;

- consist of actions and expected results
- are grouped together
- are derived from either source documentation (black) or source code (white).

The documents you need are things like test conditions, test scripts, links to source documentation. You use this documentation to capture the design effort so you don’t have to repeat it.

A key element of this process is to design an infrastructure that can provide traceability, impact analysis and crucially allow all parts of the business to participate in meaningful risk assessment.

## Test logs

- record what happened
- capture fault information to communicate to others
- give you information on risk; what has and hasn’t been tested/passed.

There are various standards you can use to get advice on documentation, for example, IEEE 829 and BS 7925-2. [See elsewhere in this issue - Ed]

Ultimately however, the most important thing is to keep focused on the benefits; ie the reasons why you are doing all this. PT

Project Name	Testing Phase	Status	Release Date	Significant Issues	Last Week Status
E11002 Distributed Order Entry	System	GREEN	8/29/02	None.	N/A
	Alpha	YELLOW	09/07/02	None.	N/A
E11002-28 - Vantage 3.5 Integration	System	GREEN	6/17/02	None	N/A
	Beta	N/A	8/26/02		N/A
	RTM	N/A	10/06/02		N/A
E11002-28 - MultiPage DPR 7.0 Integration	Beta 2	GREEN	June 17, 2002	None.	N/A
	RTM	GREEN	August 7, 2002	None.	N/A

**Status Key:**

RED Significant Quality or Schedule Issues Identified. Quality of product at unacceptable level for release. Significant concerns that schedule will not be met.

Figure 1. Example high level test status report