

# What have you let yourself in for?

For testers, moving to a different organisation which does things differently can be very challenging. Test analyst George Wilkinson advises on how to 'find your feet' quickly

How does an experienced tester assess the maturity of a new position on joining a company or a new project? Where do you start, who do you trust, how do you make a fairly rapid and unobtrusive assessment of how mature the organisation really is? And at the same time begins the learning curve that comes with a new position.

A while after joining an organisation the novelty factor will wear off. This may take a few days or even a week or two. This is a good time, when you aren't relied upon too much, to soak up the atmosphere, make contacts, network a little and of course read! Hopefully you will be instructed to read factual information about the organisation, health and safety procedures etc, but I'm really talking about doing some detective work: getting hold of and reading the organisation's requirements, functional specifications, user guides... anything that will help build up your awareness. Software documentation gives a good indication of the maturity of the development process. This should reflect on the maturity of the software test process, although the relationship may not be simple.

Questions to ask yourself include:

- Do you have requirements to work from? If so, are they testable - correct, non-ambiguous and complete - and do they make sense?
- Are the requirements at the right level for your use? If you are a system tester do you only have a set of 'high level' business requirements to work from when what you actually want is a set of system/acceptance level requirements (isn't it just wonderful when this sort of thing happens?)
- Do the functional specifications describe what the software actually does rather than how it does it, ie are they exclusive of design information? Avoiding details of the inner workings of the product makes for a good source for black-box functional testing.

I don't imagine that you will have plenty of time to do all this (if you do, let us know, so we can all apply) so I would recommend flicking through and picking requirements or sections of documents at random. This will have the added side effect of gathering information about the new product and will be viewed by others as making a good start.

## Development lifecycle

After building up some general awareness of the health of the requirements and product knowledge, start moving outwards and picking up some information about the general well-being of the software operation. This will help you broaden your picture of the organisation. Start by asking questions such as:

- What is the development lifecycle like? Is it sequential, iterative, spiral (eg RAD), other or unidentifiable?
- Do the validation and verification stages follow the waterfall model (ie no real development or integration testing stages, just one long test session at the end of development) or the V-Model (where each development stage has an appropriate test stage - see figure)
- Or is a mixture of life-cycle models used for different projects?

If it is not obvious, or they have a different way of working, try to capture it. This may involve interviewing developers, testers and project managers and drawing your own diagram or flow-chart, but it's very handy to build up a picture of the flow of events, it makes things a lot more tangible. The weaker the model the higher the likelihood that the quality is not seen as a major business concern. This is obviously a headache for the tester, but the business may not see this as a 'real' issue.

## Incident reporting

How does the organisation handle incident/defect reporting? A paper-based system? An all-singing-all-dancing tool (becoming more and more common)? Or no system at all (still not unusual)? If they have a system

examine at some of the defects raised and try to answer the following:

- How many defects there are and how well have they been written up?
- How many duplicate defects are there and how many have been rejected due to 'no-fault'?
- Are there any cluster areas of the product where more defects have been found than in others?

In general the more of these three things the worse things are. Is someone monitoring them, ie carrying out the role of a 'change control board administrator'?

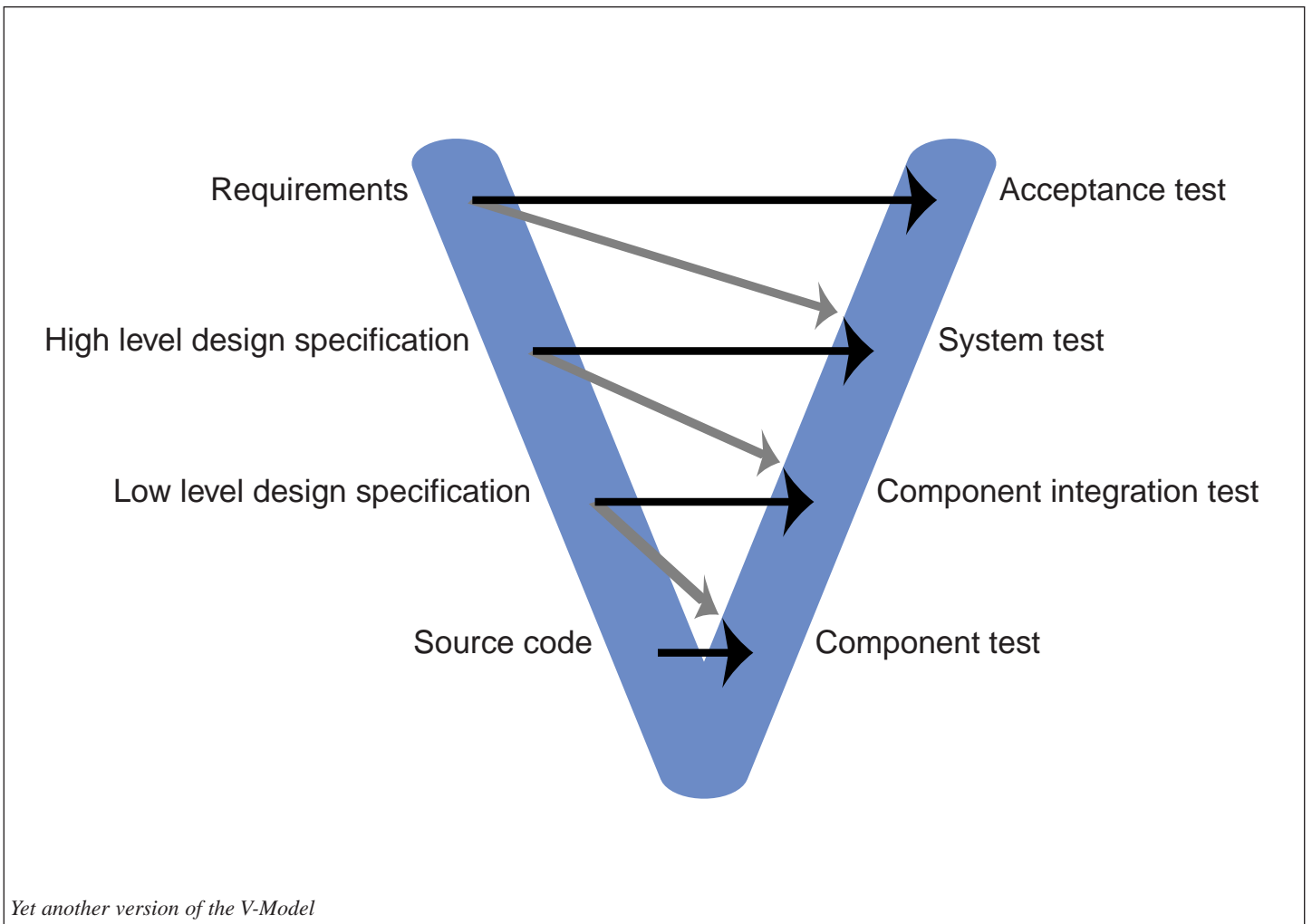
## Configuration management

What's the configuration management like? Are the developers using a configuration management tool (eg CVS, PVCS, ClearCase, SourceSafe etc)? If so, they should be entering all products required to reproduce, understand and test the software, eg source, object files, and test tools. "What?" I hear you say, "test tools?" Yes! If a software product is considered an item of test, ie we are basing test results on using tools, then the tool should be managed properly, though in practice they seldom are.

What about the testing staff? With respect to CM and testing, the things to look for here are:

- From whom do the testers actually receive the product?
- Are the testers testing the right thing?
- Do test logs trace back to test cases/procedures, and do these then trace back to the requirements?
- Is this information readily available?

All this information is required to provide a good audit trail. If tests were to be re-run for any reason, then exactly the same situation could be recreated. This is very important if we are producing either business or safety critical software and want to ensure we have found a defect successfully, but not all of the above would necessarily fit into an exploratory testing environment.



## Testing

Last, but not least, what about the testing itself?

- Does a documented strategy exist? If not, do we really know what we are doing or what we should be doing? A classic mistake made by organisations is to believe that they are doing the right thing without establishing that they are
- How is testing done? Are any specific techniques used for generating black-box test cases? Even common techniques such as equivalence partitioning and boundary value analysis often require more work to implement than people expect
- Who generates the tests and what is the attitude of the testers?
- More interestingly, what are the attitude of the developers about testing? Do they test? If so, exactly what tests do they perform?

Getting answers to questions like these from developers can be similar to extracting teeth — more so than from testers who will hopefully view you as an ally rather than an adversary — so there is a great need for diplomacy. Tell them what you are doing - trying to

build up a picture of the maturity of the *testing* within the organisation. I feel it's worth pointing out here that we are all in this game together and, sometimes, the tester with the right attitude has the upper-hand. He can make or break the relationship with the development staff, if he can be an ambassador for the test department. It's always a good thing to network a little and gain the friendship of a few on the other side of the fence, so to speak. If you remain completely independent and only talk to developers when you absolutely have to then you put up a barrier even if you don't mean to.

“getting answers to questions like these from developers can be similar to extracting teeth... there is a great need for diplomacy”

## So what if you find it's one big mess?

This may take you back to the reason why you joined the organisation in the first place. Perhaps you've come in to help improve things, or maybe you have joined quite unaware of the situation you would find. Either way, this is time to 'do' or 'get off the pot'. If you stay with the organisation, it's sure to make you a better person and a better tester, so try to look at it positively. If the task facing you seems overwhelming, try to find solace in other like-minded people, which should help in the short-term. Non testing types don't realise how lonely the job can be, so go find some tester friends - it helps.

Many testers complain that they have no direct influence over the software delivery process, and this is often the case, but don't give up without trying; most organisations have at least some managers who are willing to listen to suggestions - they tend to be the most successful ones. It seems likely that more and more organisations will be implementing process improvement and this may represent a long-awaited chance to be heard for many.

Whatever state you consider things to be in, try to keep an open mind. Don't confuse 'different from my last organisation' with 'bad'. Finally, don't forget to do everything you can to enjoy the ride!

PT