Where next for testing standards?

Including articles by:
Helen Davidson IDBS  John Kent Simply Testing  Graham Thomas Badgerscroft
Isabel Evans Testing Solutions Group  Michiel van der Voort BCS  Geoff Quentin
Load Testing for all your web applications

NeoLoad

Test your web application's performance easily and deploy with confidence!

NeoLoad, load testing solution, ensures better efficiency to perform your tests faster, while providing pertinent analyses and full support for all new technologies.

Test your Web application's performance easily and ensure trouble-free deployment thanks to NeoLoad!

- HTTP(S)
- Adobe Flex
- AJAX & RIA
- Push technologies
- RTMP
- Silverlight
- Java Serialization
- GWT
- Oracle Forms
- SOAP

More details and free trial on www.neotys.com

NEOTYS
Make sure it works!
Where next for testing standards?

What do testers want from testing standards? *Usability.*

Academic prowess and professional rigour are essential and admired, but justified only if their end product can be used to advantage by a sufficient number of testers.

The way forward is to make standards that more testers can use.

The Professional Tester Reader’s Award: help fellow testers find excellence

People that do excellent work to help others improve software quality deserve recognition, and PT’s readers are ideally placed to identify them. So we are inaugurating the Professional Tester Reader’s Award.

Yes that apostrophe is in the right place! Winners will be chosen by individual readers. If your testing has been served well please tell us about it. Any individual or organization can be nominated. To confer an award we need detail: saying your nominee was good is not enough. Please explain what was achieved and how.

If you have delivered something really special for your customer, ask them to nominate you.

In the next issue: testing across boundaries

Good testing is objective. Therefore it is not affected by differences unconnected with testing, for example those of personal origin, culture or ability. But testing is not yet objective enough, so these things do affect it. We need ways to work together despite our differences, to help us achieve our objectives in both the short (better testing) and long (objective testing unaffected by irrelevant boundaries) terms. In the next issue we’ll learn about some ways.

Edward Bishop
Editor
ISO 9001 7.x

by Helen Davidson

Applying generic “quality” standards to testing can appear complicated, perhaps because it is itself concerned with quality. But understanding what those standards mean to testing might lead to valuable improvement.

Helen Davidson explores ISO 9001 in the context of testing

IDBS, which produces software for research and development organizations, is an ISO 9001 certified company. As a test team lead there I have experienced several certification audits and many customer audits. During all of them, when talking to me, the auditors referred to clause 7.3.6 Conduct design and development validations. But I think more of 9001 than this applies to software testing: if testing is a service, the whole of section 7 Product realization becomes relevant to it.


I asked various software people who expressed quite strong but differing views. I’d like to be able to say that the testers among them agreed with one another, but they did not. I believe their opinion related to their organization’s view and treatment of testing: testers who felt the test function within which they worked was able to act independently found it easier to view it as a service. Perhaps then making testing more independent makes it resemble a service more, or perhaps treating it as a service can help to make it more independent. Many sources, for example testing syllabuses such as ISTQB Certified Tester Foundation Level, identify test independence as desirable. And outsourced testing is a service by definition.

I am lucky that independent testing is valued in my current project and that I am in direct ownership of the approach and activities. I definitely feel that the test teams are independent and provide a service. So, would applying section 7 help us to define or improve our test process and the deliverables required for...
it? Or would it just generate unnecessary paperwork? To try and find out, I studied the section in depth, attended two relevant training courses, then set about comparing our process with section 7 looking for gaps. Where I found them, I considered what we would need to do to achieve compliance.

**Clause 7.1 Planning of product realization**

TickIT, a guide to interpreting the requirements of 9001 for the software industry (see [http://tickit.org](http://tickit.org)), addresses this with its "quality plan", a description of how the product or service provided is to be developed focusing on project-specific decisions: methods, tools and techniques. When the service is software testing, this is what most testers call the test strategy. Table 1 compares the quality plan with the test strategy standard we use at IDBS.

<table>
<thead>
<tr>
<th>ISO 9001 – TickIT quality plan</th>
<th>Test strategy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer specified and implied quality requirements</td>
<td>Test objectives</td>
</tr>
<tr>
<td>Quality requirement for the project</td>
<td></td>
</tr>
<tr>
<td>Key features of the product and/or service that are likely to impact upon quality</td>
<td>Lists of project and product risks or a reference to the risk register and the prioritisation of requirements, objectives or work items</td>
</tr>
<tr>
<td>Specific risks</td>
<td>Lists of project and product risks or a reference to the risk register</td>
</tr>
<tr>
<td>References to quality system procedures to be used</td>
<td>References to test (or other) procedures to be used, eg test case review</td>
</tr>
<tr>
<td>Methods, tools etc</td>
<td>List of test tools (eg test management tool) and/or test methodologies (eg test design techniques)</td>
</tr>
<tr>
<td>Relevant statutory and regulatory requirements</td>
<td>For example US Food and Drug Administration guideline 21 CFR Part 11 (varies with sector)</td>
</tr>
<tr>
<td>Verification and validation strategy</td>
<td>Test specification and test case review</td>
</tr>
<tr>
<td>Exit criteria for procedures</td>
<td>Entry and exit criteria for phases of software testing</td>
</tr>
<tr>
<td>Deviations and in new initiatives to standard processes</td>
<td>Processes that are not being followed and details of why; details of initiatives being trialled</td>
</tr>
</tbody>
</table>

Table 1: Quality planning elements in a test strategy

**7.2 Customer-related processes**

To move toward compliance first the customer must be identified. The test function can be considered as providing a service to developers, business analysts, product owners or the software buyer depending upon the current objective of testing, so the customer varies with sector, organization, context, testing phase and testing activity.

Perhaps the best approach is to consider the information produced by testing. Whoever receives and uses that, whether to correct defects or inform decision making, is the customer. For compliance, the test service must identify and review the customer's requirements and communicate with the customer. For a test service that is achieved by reporting, so the reporting objectives must be defined, in either or both of the project test strategy or organizational test policy. The reporting frequency and mechanism must be defined in the 9001 "Quality Management System". Our test team collaborates with its customers on the test strategy using Microsoft SharePoint: this makes providing evidence of communication to auditors easy.

**7.3 Design and development**

It's important to keep sight of the fact that in the current context this is about planning, designing and developing the testing, not the software being produced. 7.3 requires that inputs are defined and outputs provided in a form that enables them to be verified against the design. Any test process that documents its strategy and plans effectively should achieve this.

The subclauses go on to require that systematic reviews, verification and validation of the design and development are performed. It may be argued that for a testing service the reviews can be of its outputs, eg test documentation. If so, this activity can also achieve verification of these against the input requirements. If the customer as identified for clause 7.2 takes part, these reviews might be able to achieve validation of the testing too.

Finally, 7.3 requires that changes to design and development are recorded, reviewed, verified and validated. It might be addressed using configuration management of the inputs and outputs,
If IT quality matters to you, you need Professional Tester, the original and best journal for software testers.

Read the latest testing news and articles and subscribe to the digital magazine free at professionaltester.com
but a better interpretation is probably that it indicates the use of a defined mechanism for test process improvement.

7.4 Purchasing
This is a hard one. It requires evidence of a formal selection, evaluation and verification process used to choose the test tools being used. If that took place some time ago, or in fact the process used was not sufficiently systematic, the only solution might be to re-specify, re-evaluate and re-verify the tools, plus alternatives, to justify (or change) past choices.

The clause could apply also to the purchase of consultancy and outsourced testing services.

7.5 Production and service provision
To meet the requirements of this it is necessary to provide evidence that the test service is provided under controlled conditions. The definition of these in 7.5 seems to align well with the test policies, strategies and plans, monitoring and measurement of test effectiveness, and configuration management mechanisms typically used by a test organization. In that case, the required verification of these can be achieved by internal audit. The requirement to exercise care with customer property is probably met by prevailing data protection and other governance standards.

It also requires that the processes are verified and in most organizations monitoring of this adherence is done through internal audits that demonstrate compliance with the described processes and allow correction of any deviations.

7.6 Control of monitoring and measuring devices
This seems to refer to the methods used for the monitoring and measurement of testing required by 7.5. Calibrating these and safeguarding them from incorrect adjustments, damage or deterioration, or even defining what that means, would be a challenge: but it may not be necessary for compliance because the clause states that it need be done only “where necessary to ensure valid results”. A periodical examination of the reporting mechanisms, perhaps including test runs with invented input data, might be considered sufficient.

Can test organizations achieve compliance with 9001 and is it worthwhile?
It is possible: the many certified outsourced testing service providers have proved that. So do their auditors follow similar reasoning to mine about section 7? Could they audit an internal test provision in the same way? It may be as well for all organizations that test software to be ready in case they do.

But compliance does not mean the test process is effective or the software products are good. I think the best way to derive benefit from 9001 and other non-testing standards is to consider what they are trying to achieve and how it applies to what we as testers understand to be good testing practice: to help us understand why our process is as it is, buy into it fully, and improve it continually.

As I have tried to show in this article, a sequential test process seems to align reasonably well with both the letter and spirit of 9001 section 7. What about testing as it is done to support an “agile” development approach? Perhaps considering that in the same way would help to improve it too.

Helen Davidson is test team lead at IDBS (http://idbs.com) which provides advanced software solutions to research and development organizations, including those working in pharmaceuticals, agrochemicals, animal health, cosmetics, petrochemicals, forensics, manufacturing, academic research and government, worldwide.
To define the things used to test, describe the relationships between them.

The things we use to test are poorly defined. Glossaries and vocabularies define terminology but not the relationships between the entities defined. Standards and syllabuses imply some of those relationships but do not define them. IEEE 829 Standard for Software Test Documentation includes a diagram showing relationships between the documents it defines, but not with other related entities. Other standards and sources describe test process, methods and techniques, but I believe any description that does not include definitions of all the entities it involves and the relationship between them is incomplete. That full definition – an entity model for testing – would make process, methods and techniques easier to define and understand and therefore to improve.

Developing the model
The model should describe all the possible relationships, not be limited to ones based on common ways of arranging test artifacts and deliverables. That would impair its usefulness as a tool for comparing and evolving different ways of working. To identify a relationship we will (i) cite or envisage examples of it in practice, then (ii) formulate generic statements about it. Regarding the definitions of the entities themselves, for simplicity in this illustration I have used definitions I assume to be in common use and/or those given in the glossaries and vocabularies familiar to testers. Using different definitions would lead to different relationships; a complete entity model will require unambiguous definitions of the terms it includes.

Specification item and test condition
Consider the following specification item,

SI1: Only an administrator can delete a user

This might cause a tester to derive two test conditions:

TC1: An administrator can delete a user
TC2: A user cannot delete another user

And if there were more privilege levels, there would be more TCs. So we can assert that

A specification item can give rise to many test conditions

In data modelling terms, there is a one-to-many relationship between the entities.

However this example is too simple.
Suppose another specification item exists:

SI50: A user attached to a current project cannot be deleted

This might create the test conditions:

TC53: A user attached to a current project cannot be deleted
TC54: A user not attached to a current project can be deleted

The real meaning of both of these is qualified by SI1, showing that the reverse relationship exists too:

A test condition can be related to many specification items

This can be depicted as in figure 1.

Test condition and test case
The reasoning for these two entities is similar: a test condition gives rise to many test cases, but each of those test cases often satisfies other test conditions as well.

So the relationship is again many-to-many.

Test case and test script

A test script can contain many test cases.
A test case can appear in many test scripts
Test script and test execution schedule

A test execution schedule can contain many test scripts. A test script can be included in many test execution schedules.

Test execution schedule and test results log

If each test execution schedule is unique, in other words a different version of it is created for each time the set of test procedures it includes is to be run:

A test execution schedule can be referred to by only one test results log (i)

But if a schedule uses relative dates, that is the same schedule can be run more than once (eg for repeated test cycles and regression testing), then:

A test execution schedule can be referred to by many test results logs (ii)

The reverse relationship depends on the nature of a test results log. If case (i) above is true, and if a new log is created each time a schedule is run:

A test results log can contain results from running only one test execution schedule only once (iii)

But if a log can be a compilation of results from multiple schedules and runs:

A test results log can contain results from running many test execution schedules once each

If case (ii) above is true, depending on the nature of the log, the relationship is either:

A test results log can contain results from running many test execution schedules each many times

A third possibility is to maintain a log corresponding to each schedule and append new results to it each time that schedule is run. In that case:

A test results log can contain results from only one test execution schedule run many times

Figure 1: a specification item can give rise to many test conditions and a test condition can be related to many specification items

Figure 2: part of an entity model for testing

Figure 3: test execution without scripting

Figure 4: exploratory testing

Figure 5: HP Quality Center
Figure 2 depicts all the relationships identified so far, assuming cases (i) and (iii) above are true.

This model is far from complete. Other possibilities, for example unscheduled execution of tests (and therefore creation of results logs), need to be analysed and added as relationships. Obvious entities to be added include test items, test plans and incident reports.

Example variations of the model
Suppose the person executing the tests has good understanding of the SUT and strong business knowledge. It can be argued that he or she can execute test conditions directly and further analysis and scripting is unnecessary (figure 3) and that this is a way to improve coverage by identifying and testing more test conditions.

Exploratory testing is defined by some as “concurrent test design, test execution, test logging and learning”. This could be taken to mean there are no test conditions or script and therefore no test schedule prior to execution. However specification items must exist, even if only in the mind of the tester: otherwise the activity is just exploration and not testing at all.

When an incident is found and reported it is usual to include instructions on how to reproduce it, and these could be considered a test script. So:

*An incident report refers to one test script*

And

*A test script describes how to reproduce one incident*

This is depicted in figure 4.

The way HP Quality Center works can be depicted as an entity model (figure 5).

Using the model
I suggest that standards dealing with testing processes and terminology should include an entity model to complete definition of the products and deliverables to which they refer.

Even more importantly, I hope that testers will find that applying this analysis method to the process they use helps them to gain better understanding and definition of that process, and useful ideas for improving it.

John Kent is managing director of Simply Testing Ltd (http://simplytesting.com).

This article is heavily abridged and adapted from a paper presented at EuroSTAR 2008 (see http://www.eurostarconferences.com) available at http://simplytesting.com/Downloads/ent.pdf which drew on discussions with members of the Software Testing Retreat to whom the author is indebted.
The Standards Working Party of the British Computer Society Specialist Interest Group in Software Testing (“BCS SIGiST SWP”) was formed around 1995 and with help from many other people created BS 7925:1998, considered by many to be the most useful testing standard published so far. The work of the SWP continues, next towards a standard for non-functional testing. All are invited to contribute. Its output is free to use and is published, with progress reports, at http://testingstandards.co.uk.

The development approach: defining what is to be tested

At the start of the millennium the SWP adopted a domain-based approach to developing guidelines for the use of non-functional test case design techniques. This enables examples to be designed and presented in the context of the kind of system to which they are most relevant, making them easier to interpret and use. When the examples are mature, they will be used as the basis for defining standard approaches and techniques. The domains chosen are:

- database
- real-time
- embedded
- management information
- e-business
- desktop
- safety critical
- process control
- scientific
- graphics
- knowledge based

Other domains can be identified but this simple and pragmatic division has allowed us to discuss and classify example software products. Any given product, and therefore the examples, may include elements belonging to more than one domain. For example, we use the “Blankchester City web site” to demonstrate testing of a system that includes elements belonging to the database, real-time, e-business and desktop domains.

Chaos, order, then chaos again: defining what is to be tested for

There was intense discussion about the non-functional areas to be chosen: unsurprisingly, because no standard approach existed at the time. The resulting list is shown in table 1. It and the list of domains form the row and column identifiers of an interactive domain-attribute matrix (see: http://testingstandards.co.uk/domain-technique_matrix.htm) used to navigate

As software ubiquity grows non-functional testing continues to become more vital yet no standard for it yet exists

Graham Thomas and Isabel Evans of the BCS SIGiST SWP explain what has been done, what remains and how you can help
the examples of using the technique(s) recommended for testing any of the attributes in any of the domains.

Since we started work, attempts to classify functional and non-functional attributes have been published by others, including in the ISO/IEC 25000 series (table 2) and the ISTQB Certified Tester Advanced Level Syllabus (table 3). ISO/IEC 25000 classifies the attributes of software to support requirements definition and measurement of the attributes but does not describe techniques. The ISTQB Advanced syllabus describes functional and non-functional attributes in a testing context, and lists some techniques. Neither provides examples of techniques applied to non-functional attributes, although testing training covering either would require such examples. Both have significant differences from our classification and from each other, different words are used to mean the same thing, and sometimes the same word is used to mean different things.

Some sources treat some attributes, for example usability, as functional, and some as non-functional. Others bypass this dichotomy by referring simply to “software qualities”, “software characteristics” etc. It should be noted that such divisions owe as much to organizational and individual quirks as to any intrinsic meaning of terms used. However the examples referenced by the domain-attribute matrix are valid whichever classification is used: they include the use of functional techniques defined in BS 7925-2 for testing attributes the SWP classifies as functional as well as the new techniques we consider non-functional.

Table 4 shows an attempt to map different terms used to describe similar attributes to the SWP list of attributes and hence to the attribute-domain matrix, which can be traced to their sources in tables 1-3 using the colour coding.

Your knowledge and experience is needed to shape the new standard

More discussion is needed about standardizing terminology, and more work is required on the domain-attribute matrix to reflect new and emerging standards. Some domains and attributes, and many attribute-domain pairs, have not yet been addressed. We are seeking volunteers to help rework the matrix and complete the examples.
<table>
<thead>
<tr>
<th>SWP</th>
<th>ISO/IEC 25000</th>
<th>ISTQBA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reliability</td>
<td>Reliability: maturity; availability; fault</td>
<td>Reliability (plus robustness,</td>
</tr>
<tr>
<td>Recovery</td>
<td>tolerance; recoverability</td>
<td>recoverability, failover)</td>
</tr>
<tr>
<td>Disaster recovery</td>
<td>Safety: commercial damage; operator</td>
<td></td>
</tr>
<tr>
<td></td>
<td>health &amp; safety; public health &amp; safety;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>environmental harm</td>
<td></td>
</tr>
<tr>
<td>Usability</td>
<td>Operability: appropriateness; recognisability;</td>
<td>Usability</td>
</tr>
<tr>
<td></td>
<td>learnability; ease of use; attractiveness;</td>
<td>Accessibility</td>
</tr>
<tr>
<td></td>
<td>technical accessibility</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Usability: effectiveness; efficiency;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>satisfaction</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Flexibility: context conformity; context</td>
<td></td>
</tr>
<tr>
<td></td>
<td>extendibility; accessibility</td>
<td></td>
</tr>
<tr>
<td>Maintainability</td>
<td>Maintainability: modularity; reusability;</td>
<td>Maintainability</td>
</tr>
<tr>
<td></td>
<td>analysability; changeability; modification</td>
<td></td>
</tr>
<tr>
<td></td>
<td>stability; testability</td>
<td></td>
</tr>
<tr>
<td>Portability</td>
<td>Portability: adaptability; installability;</td>
<td>Portability (installability, compatibility,</td>
</tr>
<tr>
<td></td>
<td>replaceability</td>
<td>adaptability, replaceability)</td>
</tr>
<tr>
<td>Performance/stress</td>
<td>Performance efficiency: time behaviour;</td>
<td>Efficiency (performance, load, stress,</td>
</tr>
<tr>
<td></td>
<td>resource utilization</td>
<td>scalability)</td>
</tr>
<tr>
<td>Security</td>
<td>Security: confidentiality; integrity; non-</td>
<td>Functional security</td>
</tr>
<tr>
<td></td>
<td>repudiation; accountability; authenticity</td>
<td>Technical security</td>
</tr>
<tr>
<td>Interoperability</td>
<td>Compatibility: co-existence; interoperability</td>
<td>Interoperability (plus configuration)</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Portability (installability, compatibility,</td>
<td>Portability (installability, compatibility,</td>
</tr>
<tr>
<td>Installability</td>
<td>adaptability, replaceability)</td>
<td>adaptability, replaceability)</td>
</tr>
<tr>
<td>Configuration</td>
<td>Conversion</td>
<td></td>
</tr>
<tr>
<td>Procedure</td>
<td>Procedure</td>
<td></td>
</tr>
<tr>
<td>NOT COVERED</td>
<td>Functional suitability: appropriateness;</td>
<td>Accuracy and suitability</td>
</tr>
<tr>
<td></td>
<td>accuracy</td>
<td></td>
</tr>
</tbody>
</table>

Table 4: mapping of the various terms used to refer to software attributes

Graham Thomas (see http://badgerscroft.com) is an independent consultant, programme test manager and testing change agent, contributing author to software testing books, and frequent conference speaker. He is currently secretary of the BCS SIG/IST SWP.

Isabel Evans FBCS CITP (see http://testing-solutions.com) is a published author of books about software testing and quality, a testing practitioner, consultant, teacher and speaker, and contributor as author and reviewer to standards and syllabuses for software testing including the work of the SWP.

Professional Tester urges its readers to consider taking part in this work which should lead to a valuable, practical standard needed by software testers and beneficial to software users.
Where next for testing standards?

**DO-178C + UML**

Rikkert van Erp reports from Düsseldorf

The ignite 2010 conference in Germany featured several presentations about emerging standards.

The ignite conference, the new name for what many PT readers will know as ICSTEST or SQC, is still the most important one for testers in Europe despite growing competition. It took place in Düsseldorf over three days at the end of April with nearly 70 presentations plus workshops and around 40 exhibitors.

Peter Heller of Airbus Operations and Sven Nordhoff of SQS discussed DO-178C, the soon to be published fourth version of DO-178 Software Considerations in Airborne Systems and Equipment Certification. PT readers may have heard of this standard on training courses; as one would expect, it is the result of an immense amount of painstaking work and considered a superb example. Many of the test design techniques used in all sectors were developed in the safety-critical industries and it’s easy to argue that should be extended to life cycle models, activities and evidence for software certification. The structure of the long process of documentation and meetings being used to develop and reach consensus on the new version is interesting, if not particularly relevant to those who will use it or try to learn from it: more exciting to testers in general was to see how DO-178 relates to other standards that are less well known outside avionics. DO-278A Guideline for Communication, Navigation, Surveillance, and Air Traffic Management Systems Software Integrity Assurance and ARP4761 Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment seem well worth a look.

The next presentation, by Jan-Hendrik Boelens of Eurocopter Group, was more technical and accessible. It introduced four technology-specific supplements to DO-178C: Model-Based Development, Object-Oriented, Formal Methods and Tool Qualification. The use of all of these is challenged and sensible constraints are placed upon it by the standard. For example: all elements in a model must be classified as defining, or not defining, requirements, and the former group must be subject to strict traceability just as requirements themselves are; all reused and inherited code must have defined requirements; a mathematical proof can replace some testing provided the requirements it assures are defined mathematically; requirements of test automation tools must be defined in the same way as for the SUT. Would stipulations like this help to prevent expensive failures if enforced in testing of business systems?

The first name we at PT look for in a conference programme is Professor Harry Sneed. We’ve been privileged to report on his presentations twice before (issues 14 and 17) and they are everything we think a testing talk should be: inspiring, very practical and acidly funny. Prof Sneed’s current position is at Universität Regensburg; we say “current” because he once related how developers refused to work while he was on the premises after he detected large blocks of very rarely executed code and we think it’s likely testers will have felt the same way when he discovered the true coverage achieved by their tests. This time he demonstrated how to measure the size, complexity, completeness, consistency and quality of UML models, so Bavarian systems architects may already be sharpening knives.

Some of the presentations from Düsseldorf plus many new ones can be seen at other forthcoming ignite conferences worldwide, including in London on 4th October 2010. Visit http://ignite-conferences.com for full details.

---

**Where next for testing standards?**

The ignite 2010 conference in Germany featured several presentations about emerging standards.

The ignite conference, the new name for what many PT readers will know as ICSTEST or SQC, is still the most important one for testers in Europe despite growing competition. It took place in Düsseldorf over three days at the end of April with nearly 70 presentations plus workshops and around 40 exhibitors.

Peter Heller of Airbus Operations and Sven Nordhoff of SQS discussed DO-178C, the soon to be published fourth version of DO-178 Software Considerations in Airborne Systems and Equipment Certification. PT readers may have heard of this standard on training courses; as one would expect, it is the result of an immense amount of painstaking work and considered a superb example. Many of the test design techniques used in all sectors were developed in the safety-critical industries and it’s easy to argue that should be extended to life cycle models, activities and evidence for software certification. The structure of the long process of documentation and meetings being used to develop and reach consensus on the new version is interesting, if not particularly relevant to those who will use it or try to learn from it: more exciting to testers in general was to see how DO-178 relates to other standards that are less well known outside avionics. DO-278A Guideline for Communication, Navigation, Surveillance, and Air Traffic Management Systems Software Integrity Assurance and ARP4761 Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment seem well worth a look.

The next presentation, by Jan-Hendrik Boelens of Eurocopter Group, was more technical and accessible. It introduced four technology-specific supplements to DO-178C: Model-Based Development, Object-Oriented, Formal Methods and Tool Qualification. The use of all of these is challenged and sensible constraints are placed upon it by the standard. For example: all elements in a model must be classified as defining, or not defining, requirements, and the former group must be subject to strict traceability just as requirements themselves are; all reused and inherited code must have defined requirements; a mathematical proof can replace some testing provided the requirements it assures are defined mathematically; requirements of test automation tools must be defined in the same way as for the SUT. Would stipulations like this help to prevent expensive failures if enforced in testing of business systems?

The first name we at PT look for in a conference programme is Professor Harry Sneed. We’ve been privileged to report on his presentations twice before (issues 14 and 17) and they are everything we think a testing talk should be: inspiring, very practical and acidly funny. Prof Sneed’s current position is at Universität Regensburg; we say “current” because he once related how developers refused to work while he was on the premises after he detected large blocks of very rarely executed code and we think it’s likely testers will have felt the same way when he discovered the true coverage achieved by their tests. This time he demonstrated how to measure the size, complexity, completeness, consistency and quality of UML models, so Bavarian systems architects may already be sharpening knives.

Some of the presentations from Düsseldorf plus many new ones can be seen at other forthcoming ignite conferences worldwide, including in London on 4th October 2010. Visit http://ignite-conferences.com for full details.
Examinations leading to ISEB Practitioner Certificates are soon to be withdrawn. What does that mean for testers holding and/or seeking qualifications?

Michiel van der Voort explains the reasoning behind the decision and his view of the future.

**BCS recently announced** the withdrawal of its ISEB Practitioner Certificates in Test Analysis and Test Management and that it will offer examinations leading to the International Software Testing Qualifications Board (ISTQB) Certified Tester Advanced Level Test Analyst, Technical Test Analyst and Test Manager qualifications.

**ISEB Practitioner exams will cease in March 2011.**

Michiel van der Voort, Executive Director of BCS, The Chartered Institute for IT, answers our questions.

---

**Professional Tester: To help place the recent announcement in context, please tell us how many people so far have been awarded the:**

- ISEB Practitioner Certificate in Test Analysis
- ISEB Practitioner Certificate in Test Management
- ISTQB Certified Tester Advanced Level Test Manager qualification
- ISTQB Certified Tester Advanced Level Test Analyst qualification
- ISTQB Certified Tester Advanced Level Technical Test Analyst qualification.

— Michiel van der Voort: As of 19th April 2010: 158, 195, 2,996, 2,285 and 1,110 respectively.

**Why are the Practitioner Certificates being withdrawn?**

— We realised that two different certification tracks were developing for software testing professionals depending on whether or not they lived in the UK. Even though we are incredibly proud of our ISEB Practitioner Certificates, our decision to withdraw them is because we feel it is not in the best interest of the software testing community to have disparate competing paths and that the global standing of ISTQB needed to be acknowledged.

Withdrawing two of your best exams is not something to be undertaken lightly, but we have done it for the good of the international IT profession.

Since the ISEB Intermediate Certificate in Software Testing is not a prerequisite for ISTQB Advanced Level, do you expect continued demand for it?

— Yes. It covers the testing fundamentals to help newly qualified
Testing qualifications

testers learn the skills needed to take them to the next level, is a natural progression from the ISTQB Foundation Certificate, and provides broad coverage of the whole discipline of software testing.

The ISTQB Advanced Level qualifications suit candidates who wish to gain a much more in-depth knowledge and understanding of the separate software testing disciplines at a practical level and are aimed at people who have achieved an advanced point in their careers in software testing. I believe the Intermediate Certificate has great value also as preparation for them.

How do holders of ISEB Practitioner Certificates stand regarding ISTQB qualifications?

— The Practitioner Certificates, including those awarded against the old-style all-in-one syllabus before it was split into Test Analysis and Test Management, will count as some of the necessary prerequisites for the ISTQB Expert level when it becomes available.

ISTQB examinations are entirely multiple-choice. How do you respond to the criticism that passing such an exam is not sufficient demonstration of practical ability for a person to be designated, for example, “Advanced Level Test Manager”?

— All examinations are a snapshot of a person’s abilities, by definition. ISTQB prefers multiple-choice exams because they eliminate subjectivity in marking. BCS will be monitoring the examinations to make sure they are rigorous and valid as we do our own examinations.

Will ISTQB Expert Level also be examined by multiple-choice?

— This is yet to be decided. This agreement places BCS in a stronger position to work with ISTQB on the specification of the Expert Level, and it is likely that the experience that ISEB has gained from the Practitioner Certificates examinations will be helpful to ISTQB as that specification matures.

Is it still intended to launch the ISEB Diploma in Software Testing?

— Yes. The ISEB Diploma will be a wider qualification than the Practitioner Certificates and ISTQB Advanced qualifications, encompassing a different set of focus points for software testing professionals looking to widen their horizons.

What about training providers which are accredited or seeking accreditation to present Practitioner Certificate courses?

— We are in ongoing discussions on a case-by-case basis with all organisations involved, who we contacted as soon as agreement was reached.

Opinions differ on the relative merits of the ISEB and ISTQB syllabuses. Obviously the BCS believes the ISEB Software Testing qualifications are/were good. Is there any regret at their loss?

— In any agreement between different organisations serving similar markets, some compromise has to be made. BCS has accepted that whereas its Practitioner Certificate examinations were particularly relevant to a more mature software testing market such as that in the UK, the rest of the world and a large and growing number of UK based international organisations were moving to a single international standard. BCS hopes now to work more closely with ISTQB to develop a qualification portfolio for the benefit of the overall market.

What guarantees can you give to potential candidates, training buyers and training providers about future changes?

— Qualifications are always under development and brought up to date with development undertaken in close cooperation with the market, both the training providers and the industry. BCS is determined to be seen as a global player in IT and the ICT profession. Our Transformation programme is part of this same globalisation. We will endeavour to engage with all partners closely.

Michiel van der Voort is Executive Director of BCS, The Chartered Institute for IT (see http://bcs.org/qualifications). Interview by Edward Bishop.
The V Model according to ISO/IEC 12207

The Consolidated Testing Process (“CTP”) is based on the V model: the concept that tasks in a development process and deliverables produced during their performance can be used as basis for test material needed later in that development process.

The CTP requires that the deliverables themselves are tested. In the CTP, testing non-executable deliverables is called analytical testing. Testing executable deliverables is called empirical testing.

A version of the V model with activities as defined in ISO/IEC 12207 Software Life Cycle Processes (“12207”) is shown in figure 1.

Issue tracking

Testing may identify anomalies, ie differences between what is expected and what appears to be present. However the cause of the anomaly may be in the test itself or in a person’s understanding. In the CTP, until the nature of the difference is known, the existence of the difference is referred to as an issue.

When an issue is identified the CTP requires that it is: assigned a unique reference; recorded in such a way that it is traceable from identification to resolution and the record is maintainable throughout the life cycle; analysed, graded and reported; resolved.

One approach to grading (“classifying”) and tracking issues is described in IEEE 1044-1993 Standard Classification for Software Anomalies (“1044”).

Issue tracking during analytical testing

The CTP requires that tests are planned and designed as early as possible in the acquisition process. If this activity causes an issue to be identified, the CTP requires that its resolution includes the creation of test material to be used during empirical testing.

Identification and resolution of issues by analytical testing during specification and design processes is desirable.

Issue tracking during empirical testing

Identification of an issue during empirical testing is undesirable. It indicates failure of earlier quality control. The CTP requires that relevant action intended to improve the development process, test process or both is initiated in response to it.

If the cause of the issue is found to be in the test object and not in the test that identified the issue, the CTP requires that the test activity is repeated before resolution (“re-testing”) and after resolution (“regression testing”) to demonstrate that the difference between what is expected and what is present which enabled identification of the issue is no longer apparent.

Incident management

Identification during empirical testing of an issue whose cause is found to be in the test object and not in the test indicates that the test object has the potential to fail, perhaps delaying deployment, and requires rework and repeated test activity, perhaps delaying continuation of development and testing work. Because of these time-related effects and their implications for project management, the event of identification of such an issue may be called an “incident”.

Organizational testing strategy

The CTP requires that management...
The CTP requires that root cause analysis is carried out for every issue identified in any deliverable to determine during what development or testing process the error that caused the issue occurred. That process is then modified by incorporation of the testing activity that detected the defect, improving the process and reducing the need for empirical testing.

Section 2

The Generic Test Process

The CTP requires that the Generic Test Process ("GTP") is followed to test all test objects. A test object is any object that is of concern, ie the products of development processes such as those defined in 12207. The testing can be done using any suitable method including but not limited to those defined in BS 7925-2 "Software Component Testing ("7925") and IEEE 1028 Standard for Software Reviews ("1028") and documented for example as described in IEEE 829 Standard for Software Test Documentation ("829").

The CTP requires that processes and documentation methods are chosen for their ability to support and not to hinder the testing work. The role of a tester is not to fill in forms or adhere to development processes but to (i) attest to an attribute or deliverable being correct when it is correct; (ii) detect and record in a way that can be analysed defects that make an attribute or deliverable incorrect.

The GTP is designed to support management by making it easy to understand progress via clear, meaningful, continuous reporting. It consists of seven tasks:

1. **GTP1 Test status reporting**
2. **GTP2 Create the test strategy**
3. **GTP3 Create the test plan**
4. **GTP4 Perform test analysis**
5. **GTP5 Design test(s)**
6. **GTP6 Schedule execution of test(s)**
7. **GTP7 Run test(s)**

**The Test Management Process**

The Test Management Process ("TMP") consists of the following tasks: assigning roles and responsibilities; checking that tests are correct; execution of tests; maintenance of test material including when requirements change.

**Business driven testing**

The CTP requires that risks are identified, documented, defined, quantified and prioritized, that tests intended to support the containment of each identified risk are developed, and that those tests are executed in an order derived from the risk prioritization.

**Testing driven by business benefits**

The CTP requires that all functions and all non-functional attributes ("NFA’s") identified in the requirements for the product are graded in terms of business benefit, that tests to show that each function and NFA is compliant with the completion criteria are developed, and that those tests are executed in an order derived from the business benefit grading.

**Process improvement**

The CTP requires that metrics are gathered from all testing activities (analytical and empirical) and used to identify development and testing processes that may benefit from assessment of their effectiveness. All processes are open for improvement at all times.

supports an organizational testing strategy which identifies: all the quality control (QC) points; who is to carry out the testing at each QC point; the expected compliance requirements (internal to the development and supply processes); the reporting structure; overall system quality requirements; risk containment tasks; external compliance requirements (such as legislation or standards).

The organizational testing strategy governs the testing including the types of testing to be done and the balance between the effort spent on each of them, test environments, scope (with test objectives and a coverage requirement), tools, maintenance of test material, expected timeframes, reporting, control processes and overall administration. The CTP requires that the organizational testing strategy is supported by appropriate management policy statements and compliant with applicable standards.

The CTP requires that root cause analysis is carried out for every issue identified in any deliverable to determine during what development or testing process the error that caused the issue occurred. That process is then modified by incorporation of the testing activity that detected the defect, improving the process and reducing the need for empirical testing.

Section 2

The Generic Test Process

The CTP requires that the Generic Test Process ("GTP") is followed to test all test objects. A test object is any object that is of concern, ie the products of development processes such as those defined in 12207. The testing can be done using any suitable method including but not limited to those defined in BS 7925-2 "Software Component Testing ("7925") and IEEE 1028 Standard for Software Reviews ("1028") and documented for example as described in IEEE 829 Standard for Software Test Documentation ("829").

The CTP requires that processes and documentation methods are chosen for their ability to support and not to hinder the testing work. The role of a tester is not to fill in forms or adhere to development processes but to (i) attest to an attribute or deliverable being correct when it is correct; (ii) detect and record in a way that can be analysed defects that make an attribute or deliverable incorrect.

The GTP is designed to support management by making it easy to understand progress via clear, meaningful, continuous reporting. It consists of seven tasks:

1. **GTP1 Test status reporting**
2. **GTP2 Create the test strategy**
3. **GTP3 Create the test plan**
4. **GTP4 Perform test analysis**
5. **GTP5 Design test(s)**
6. **GTP6 Schedule execution of test(s)**
7. **GTP7 Run test(s)**

**The Test Management Process**

The Test Management Process ("TMP") consists of the following tasks: assigning roles and responsibilities; checking that tests are correct; execution of tests; maintenance of test material including when requirements change.

**Business driven testing**

The CTP requires that risks are identified, documented, defined, quantified and prioritized, that tests intended to support the containment of each identified risk are developed, and that those tests are executed in an order derived from the risk prioritization.

**Testing driven by business benefits**

The CTP requires that all functions and all non-functional attributes ("NFA’s") identified in the requirements for the product are graded in terms of business benefit, that tests to show that each function and NFA is compliant with the completion criteria are developed, and that those tests are executed in an order derived from the business benefit grading.

**Process improvement**

The CTP requires that metrics are gathered from all testing activities (analytical and empirical) and used to identify development and testing processes that may benefit from assessment of their effectiveness. All processes are open for improvement at all times.
GTP1 is in two parts. The very first activities of the GTP are establishing the method of referring to the testing work to enable traceability and defining the test status reporting mechanism. Then, maintenance of traceability and reporting are performed continuously in parallel with GTP2-7, using that reference method and reporting mechanism.

The reference method must allow recursion, so that test objectives can be of multiple levels, and it is clear of which higher-level test objective each lower-level test objective forms part, and on which lower-level test objectives each higher-level test objective depends.

GTP7 refers to the running of testing methods such as reviews as well as the execution of empirical tests.

At least the following information must be recorded and maintained for every test, throughout the GTP: test reference; business objective; test object; test objective; test attribute (condition to be tested); completion (compliance) criteria; test method; test data; test setup; test schedule; test execution; test status

The business objective is recorded during GTP2, test object and test objective during GTP3, test attribute and completion criteria during GTP4, test method, test data and test setup during GTP5, test schedule during GTP6 and test execution during GTP7.

Additional information may be recorded for each test to achieve auditability but is not required by the CTP.

Test status is one of a set of indicators, for example: test reference assigned; business objectives identified; critical success indicators defined; test strategy defined; test objects identified; test objectives set; attributes defined; completion criteria specified; method identified; data created; environment and setup created; data, environment and setup checked; run date assigned; resources allocated; pre-conditions met and test ready to run; failed, re-test needed; passed, accepted with qualifications; passed, accepted

The CTP does not prescribe the status indicators; the set may be flexible and chosen to suit the work at hand and the indicators may be many and various. But the CTP requires that every status indicator defined is recorded against each test at the appropriate time: no indicator may be skipped.

By updating the status indicators every time a test activity is completed for an individual test, the tester is in a position to provide accurate reporting throughout the GTP. Management can understand progress clearly and easily by referring to the test reference, business objective, test objective and test status only.

In the next issue The Consolidated Testing Process part 3: the Generic Test Process: test strategy, planning, analysis, design, scheduling and execution

Top 5 reasons not to use a standard

1. **You can’t find it**
   Most Google results lead to Powerpoint slides about the standard designed by a chimp. What may or may not be the standard itself seems to be behind a paywall thicker than Rupert Murdoch. Someone should start a torrent site (idea © Professional Tester 2010)

2. **You can’t read it**
   If you do manage to get a copy of ISO/IEC 15504 Software Process Assessment and keep it at your bedside you’ll never need temazepam again

3. **It’s ambiguous**
   Some standards appear to have been reviewed to destruction by a committee determined to find and remove any assertion with which anyone in the world might conceivably disagree

4. **You can’t do it by yourselves**
   Some standards require an expert assessor (a bit like a school inspector) to show up twice a year, drink your coffee, tick some boxes and issue a large invoice

5. **You didn’t know it applied to you**
   The black-box techniques defined by BS 7925-2:1998 Standard for Software Component Testing are equally applicable at all testing levels. Would leaving out the word “component” have led to it being used more in the last 12 years?

Send your testing top 5 to editor@professionaltester.com. If we publish it you will receive a free subscription to the printed edition
COGNIZANT TESTING ROUNDTABLE 2010

Cognizant would like to invite you to an interactive knowledge sharing event where we will be discussing professionalization in testing.

AGENDA

<table>
<thead>
<tr>
<th>Time</th>
<th>Session</th>
</tr>
</thead>
</table>
| 14:00 - 14:15 | Registration  
Welcome, Meet & Greet                                               |
| 14:15 - 14:30 | Opening Remarks  
Future Trends in Testing & Cognizant Testing Practice Overview  
Sumithra Gomatam (Head, Cognizant Testing Services)              |
| 14:30 - 15:15 | Keynote  
Major Trends in Software Testing  
Dominique Raviart (Research Manager, Nelson Hall)  
The speaker will discuss the constraints in traditional testing methods, pros and cons of new testing offerings in the market, various contractual forms and evaluation of key selection criteria for offerings and vendors. |
| 15:15 - 15:30 | Break |
| 15:30 - 16:30 | Roundtable Discussion  
The Next Step in Test Professionalization                           |
| 16:30 - 16:45 | Closing Remarks  
Sumithra Gomatam (Head, Cognizant Testing Services)              |
| 17:00      | Cocktail & Dinner                                                     |

JUNE 10TH 2010

from 14:00 to 17:00  
Bossche Boardroom in Den Bosch  
Magistratenlaan 186  
5223MA Den Bosch  
The Netherlands

As seating is limited, sign up now to secure your presence at this interactive round table. You can register by sending an email to esther.snijders@cognizant.com

Your future, on your terms