

TickIT International

The quarterly journal of the TickIT software quality certification scheme ISSN 1354-5884

CONTENTS

EDITORIAL	PAGE 2
TESTING IN THE eXTREME PROGRAMMING (XP) PARADIGM: A CASE STUDY	PAGE 3
<i>by Marnie L. Hutcheson</i>	
RIGHT PRODUCT, RIGHT PROCESS	PAGE 11
<i>by Daniel Dresner</i>	
A REPORT ON THE BCS QUALITY SIG'S SQM 2004 CONFERENCE	PAGE 15
<i>by Margaret Ross</i>	
RUMBLINGS OF A TICKIT TRAINER	PAGE 17
<i>by Neil Martin</i>	



IT 3Q04

Recently we have carried a number of articles on Agile methods. In these articles we have always identified the importance of testing and working with the customer to establish proper functional tests before release. This, however, is not always the case – as is demonstrated by the case study from Marnie Hutcheson on the risks of not undertaking independent system functional tests. So, please ensure that testing doesn't get short-changed in any rush to embrace Agile methods.

It is some time now since the DTI-sponsored 'Towards Software Excellence' programme was launched by the NCC and cut its funding ties from government – I wonder how it is getting on? Well, I approached Daniel Dresner to give us an update on how it is faring, and what is planned for the future – he kindly obliged, but ended up taking us to a desert island!



The BCS QA Special Interest Group held their SQA INSPIRE conference earlier this year, so I asked them how it went; a short report resulted and a call for papers for next year's event – so you can get drafting.

The TickIT Scheme has had its supporters and critics through the years. It is important that we, who are part of the scheme, listen to the comments they have and respond positively.

Now, Neil Martin has been involved in TickIT training for some time and has run a TickIT Survey on his web site.

We asked him to give us a summary of the sort of findings that he was observing both from training delegates and from his surveys. It would be good to dig a little deeper, so I add my voice of encouragement to Neil's to invite readers to participate through his surveys. Maybe there are some opportunities for improvement that we can feed back to the TickIT committees ...

ALL COPY AND LETTERS TO BE SENT TO THE EDITOR, AT THE FOLLOWING ADDRESS:

TickIT Office, BSI, 389 Chiswick High Road, London W4 4AL.

Tel +44 (0)20 8996 7427 / Fax +44 (0)20 8996 7429

email: tickit@bsi-global.com

COPY ON DISK OR EMAIL PLEASE – COPY SHOULD NOT BE SENT TO THE PUBLISHER

COPY DEADLINES:

December 23 for publication January 15

March 31 for publication April 15

June 30 for publication July 15

September 30 for publication October 15

For advertising sales contact

Tina Shorter, Firm Focus, Folia, Flowers Hill, Pangbourne, Berks RG8 7BD.

Tel +44 (0)118 984 3949 Fax +44 (0)118 984 2493

email: tina@firmfocus.co.uk



Published by Firm Focus on behalf of BSI July 2004

www.firmfocus.co.uk



Testing in the eXtreme Programming (XP) Paradigm: A Case Study

By Marnie L. Hutcheson

INTRODUCTION

In order to compete in today's market, it is a business imperative to develop the web. Not only must companies move their existing products on to the web, but they also have to develop new products directly for the web. Market pressures in the global Internet environment are intense; there is a constant pressure to produce cheaper, faster, and smarter software. This trend will continue to escalate over the next few years as the world moves into mobile Internet computing.

We are all being forced to re-evaluate the processes we use to create software products. The 'heavyweight' or traditional approaches to producing quality software are largely too cumbersome and too expensive to be competitive in this environment – and they do not ensure that the end product will satisfy the users. The 'lightweight' approaches, descendants of the Rapid Application Technique, RAD, called agile approaches, are geared to be very responsive to the needs of this environment and, as a result, have had more success producing software for today's Internet. Agile approaches in general, and eXtreme Programming (XP) in particular, are focused on finding new, more efficient ways to deliver high-quality software, fast.

As budgets and schedules shrink, everyone in the development effort is feeling pressure to demonstrate the value that they add to the product. Testers, in particular, are being asked to show how they add value to the development process and how much that value is worth to the company. XP takes this situation to the extreme by asserting that testers are unnecessary and even counter-productive – that they do not add any appreciable value to the product. This article gives an account of an informal test effort conducted on a recent real-world XP project.

Besides providing a good look at how an XP project delivers, it also provides a good opportunity to measure and quantify the value that testing added to the project.

BACKGROUND ON (XP)

The Internet lends itself to small, lightweight applications by its very nature. The current trend is to write very small modular components that perform a particular task and call on them as needed. These program components may be spawned from vastly different architectures and may be called different names by

different vendors; I will call them web services. Web services are based on standards set forth by the W3C, including html, XML, SOAP and UDDI.

XP evolved in this standards-based, small application production environment. It is a 'small project' methodology; it's not known to scale well (6-12 developers is considered ideal). As such, it is ideally suited to Internet development projects. For this and several other reasons, XP has many adherents among web developers. XP is an iterative development methodology that is based on 12 basic tenets. The source for this listing is Kent Beck's book: *eXtreme Programming Explained: Embrace Change*.

1. customer is at the center of the project,
2. small releases,
3. simple design,
4. relentless testing,
5. refactoring (adjust code to improve the internal structure; make it clean, simple, remove redundancies and so on),
6. pair programming,
7. collective ownership,
8. continuous integration,
9. 40-hour work week,
10. on-site customer,
11. coding standards,
12. metaphor – development is guided by a story view of how the system will work.

The adoption of these methods is often driven from the bottom up by developers who are frustrated by the constraints placed on them by the more formal processes associated with traditional plan-driven approaches to developing software. XP has quickly become very popular in development communities as a way to maximize their efficiency and produce products that satisfy their market.

As with any new approach, there are strong points and weak points in the process. Several of each are illustrated in this case study. Before presenting the study I want to discuss some of the fundamental differences between the traditional and the agile approaches.

PLAN-DRIVEN VERSUS AGILE METHODS

Agile and plan-driven teams use different approaches to design and implement software. The two most important differences in this particular project are:

1. the difference in spending priorities,
2. the absence of testers.

Next column



The Difference in Spending Priorities

The plan-driven methodology believes in spending up front to acquire information that will be used to formulate the design, so as to be as well-informed as possible before committing to a plan. This is called *money for information* (MFI).

The agile method chooses the design early and modifies it as it goes. So budget is reserved to help adjust and absorb the changes and surprises that will certainly come. This is called *money for flexibility* (MFF).

Spending money for information, or MFI, is generally considered to be part of the plan-driven management strategy. Reserving money for flexibility, or MFF, is considered to be part of the RAD/agile management strategy (see Figure 1).

The plan-driven management is willing to spend money for information to mitigate risk. When the outcome is uncertain, as with new technologies or systems, MFI is used for unpredictable and ‘all-or-nothing’ scenarios – for example, when we don’t know actual usage patterns, or system limits. These are things that can be measured or simulated in advance of development. That way there is time to change course before the project fails.

The problem in developing web applications is that the fast pace of development and the fluid nature of the Internet create an environment with **too many unknowns**. Companies using traditional approaches typically find themselves over budget, behind schedule and – worst of all – they often fail to produce software that satisfies the users. After all, how can a static plan with a static goal hit a constantly-moving target? Too often it is the case that by the time the application is implemented and rolled out, the needs have changed and the product is not satisfactory.

The agile approach implements in small increments, produces a functioning product or feature set as fast as possible, evaluates it, refines it and iterates

through the process again – and the product evolves as it goes. A functioning product, albeit a limited one, is available from very early in the development process. By reserving funds for flexibility, the agile effort is gambling that those last-minute problems can be fixed. This flexibility was a key element of the success of the project in this case study.

The Absence of Testers

Even though item four of the XP tenets lists ‘relentless testing’ as one of the basic tenets, it does not say ‘who’ will do this testing. Apparently this item is open to interpretation. The first XP project I was involved with cast me in the double role of tester and technical writer. It worked very well, and was the subject of the paper *The Team with the Frog in Their Pond* (1999). In that project, each feature team consisted of a developer, a tester and a business partner (the customer). I was completely taken by surprise to hear that testers were ‘unnecessary’ in this 401(k) web application XP project. All testing was done by the developers and the customer. There was no end-user testing, no integration testing, nor was there any process flow or structural testing performed until I began working on the project.

THE 401(K) WEB PROJECT

This was a web-based application that would provide 401(k) retirement plans for self-employed, small-business people. Up until this new product line, the benefits of a 401(k) retirement plan had been the exclusive province of large businesses. Not only had there not been any opportunity for self-employed persons to have a 401(k) program, but there was also a significant number of now self-employed persons whose 401(k) retirement plans had been frozen in the 401(k) plan of their last major employer. People with their 401(k) plans in this situation couldn’t contribute to them or borrow from them. These assets were effectively una-

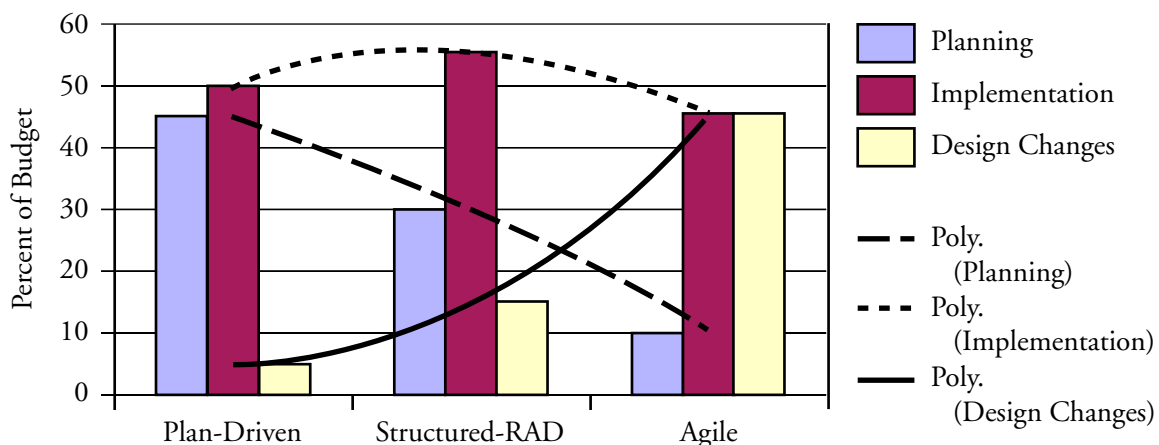


Figure 1: Development Methods and Their Spending Priorities

available to their owners until they rolled them into an IRA, which could accept contributions but from which they could not borrow. The other option was to close the fund and remove their money, and pay a significant penalty and income tax on the amount.

This new product would allow self-employed people to set up and administer their own 401(k) plan. It would also allow them to transfer in their lame duck 401(k) holdings, contribute to it, and borrow against it. The response to the plan offering was expected to be huge.

The new product was created for the Fiduciary Trust (the customer) by an Internet Service Vendor (ISV) who also hosted the application as the Application Service Provider (ASP). The end user is the individual who subscribes to the plan.

TESTING CRITERIA

I became involved with this project as a potential customer, not as a hired tester. As a self-employed web consultant, the web-based 401(k) product was very appealing to me. But I switched from customer to 'professional tester' before I got past the first screen. Since it was my money, and there were no other offerings of this type available, I decided to pursue the opportunity to test this application and report my findings in order to see what would happen.

I was not given any instructions or description of how the process worked. I approached the project armed only with my ID, my newly issued password, my Internet-connected PC, and a blank document open in Word. I used the MIT's test methods, a risk-based test methodology described in my new book *Software Testing Fundamentals*. Since I wasn't given any specific requirements, I adopted the fundamental risk requirements for all fiduciary dealings:

- No money can be created or destroyed (or lost, or prematurely withdrawn).

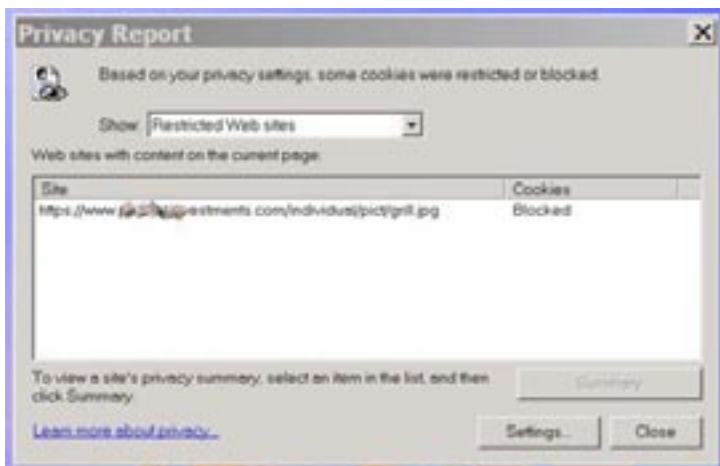


Figure 2: The Logon Failure
Next column

The main risk, if any of these conditions should occur, can include financial instability, audits by government agencies, criminal proceedings, and high-stakes litigation. I used this one requirement and these risks to rank the bugs that I found.

ERRORS REPORTED

WHILE TESTING PLAN ENROLMENT

My first test was to log on and enrol in the plan. Within a few hours I had identified three fatal errors in the application and two fatal errors in the process. In four hours of testing, I logged 20 issues. Of these, 15 required a call to customer service (CS) to resolve so that I could continue; five were of a nature that would cause the customer to lose confidence in the institution. I found all of these issues simply trying to enrol in the program.

1. Could not log in on www.maincompany.com/individual401k – cookie error on redirect

Problem description: The main corporate web site was using cookies to redirect the subscriber to the application which was hosted by the ASP/developers. These cookies were rejected by both the Internet Explorer versions 6 and 5 browsers when set at the normal default security settings. These two browsers represent 70-80 percent of all web traffic to all the web sites hosted by Ideva. That means that probably **70-80 percent of all potential users would fail to log on – every time**. The disclaimer on the logon window states that the application is optimized for the Netscape browser. Statistics from the last half of 2002 show that all versions of Netscape amounted to less than 10% of all traffic to these sites.

The logon failed to complete because it violated the security settings in the browser, thus creating an endless logon loop where the logon succeeded but no other page could be displayed. This caused the system to redisplay the logon page. However, no error message was displayed. The only clue to the actual events was a small icon displayed in the bottom border of the browser – which was not an obvious source of information. Some exploration of the browser policy and security settings revealed the actual problem (see Figure 2). Non-technical users would have been completely stymied – and would have failed to log on. See the sidebar on first time user failures in Calculating the Value of Testing on page 7.

The workaround: the ASP was not using these cookies, so it was possible to navigate directly to their logon URL and log on to the application.

Note: Three months later, the problem still existed on the main web site. The

lead developer at the ISV tried repeatedly to get the company's IT personnel to fix the problem, but with no success. Apparently the core issue was related to support for the Netscape browser over IE.

2. Plan page interface problems – see Figure 3

- 'GUST' is not explained in paper instructions but is clarified online.
- Several fields are too short to display all the required input information, that is, the date field in the screen pix – notice the error message about the year needing to be within 100 years of today's date.
- Typeface too small for age >40 – the font size was set to 8 points, and could not be changed by the user.
- Bank ABA # in the online interface is called ACH R/T on bank checks (bottom of screen), required a call to CS to get the translation.
- No way off page – there is no 'continue' or 'next' or 'back' button on this page.

3. Plan setup page function and capability problems

- I downloaded the 61-page PDF and the machine locked up during printing. This error repeated

several times – it seems to be contention between Acrobat Reader and Outlook. Later analysis showed that the lock-up was due to malformed coding introduced by the application logic as it encoded the agreement into PDF format. No agreement created with these errors could ever print. Further, both the document – stored in memory – and the online session were lost when the machine locked up.

- After rebooting and coming back to the site, it was no longer possible to get back to the plan setup page, so could not get another copy of the plan agreement.
- After the lock up and reboot, the next time I logged on to the plan setup, the system took me to the census page. There was no way to get back to the plan setup page to complete the setup process, and no indication that there would be an opportunity to agree later in the process. The system took me to a new set of pages, and the outcome of the agreement was unknown.
- **The system had erroneously decided that I had agreed to the terms of the plan – but I had never succeeded in getting to the agreement page, because of the problems with PDF locking up the PC and**

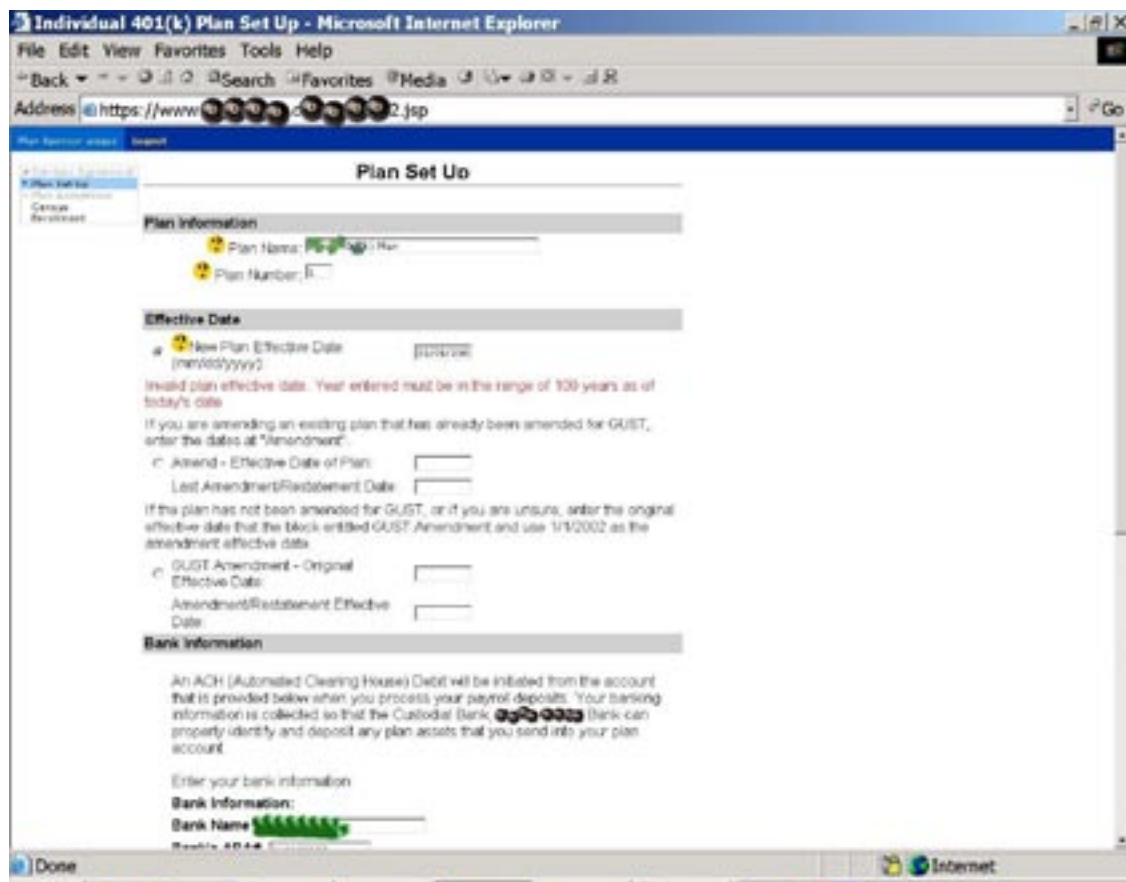


Figure 3: The Plan Setup Page

prematurely ending the session with the plan setup application.

- The plan document is 61 pages – not likely that anyone is going to read it and agree in the same session – but there is no way to re-enter the setup process.

4. Enrolment

During enrolment, allocation selection is conducted using percentages, but the allocation field doesn't accept the % sign. The error message talks about 'whole numbers' only – user confusion will occur.

5. I entered in my beneficiaries' social security numbers with dashes between number groups

The system (gave no instructions for or against, and no data example was presented) inserted more dashes when I submitted the page. The resulting social security number was corrupted.

- Next, I couldn't edit the beneficiaries' social security number. The solution was to delete and re-add the beneficiary.
- There is no way off this page.

6. Information windows don't close and don't have a button or link ...

... and they don't come forward when a new topic is selected. They remain behind the browser, so it looks like the help function is unresponsive or absent.

7. There is no closure on the plan setup process

After census, you can enrol, then there is no wrap up, and since I never agreed to anything, the process has failed. I can't find a way back to log on as a plan owner either. So how can I administer the plan? (The top menu has only two options, Participant access and Log out.) The options in the left menu are now all geared to the employee, not the employer. The developer tells me that I should have an option in the top menu to switch to the administrative role.

8. The best system issue:

Both the company site and the ASP site disappeared from the Domain Name Server (DNS) in the second week of testing.

9. The best process flow issue:

The bank in charge of processing the funds, both contributions and rollovers, lost the rollover check, because they didn't know how to process it.

10. Miscellaneous

The administrator's instructions were completely out of date and actually belonged to a different product, so following the instructions in the administrator's guide caused serious errors in the user's plan.

Next column

RESULTS REPORTING

Most of the issues were first reported over the phone, but I also prepared a comprehensive written report, which was widely circulated among the developers and the agents. This document was used as the agenda in several conference calls between the developers, customer service, and me. All the reports that I submitted were also circulated to the project team at the fiduciary trust.

I created an inventory of the features that I tested; the logic flow map of enrolment (not presented in this article), and a bug report complete with pictures. The logic flow map clearly showed two of the fatal errors. These could have been detected in a design review. The report also contained my cost/savings analysis attributable to the test effort. It was the cost/savings estimate that got the most attention.

CALCULATING THE VALUE OF TESTING

There were two areas where the value of testing could be demonstrated in this effort. The first is in terms of the cost to the CS departments for helping each customer through the enrolment process. (This neglects the rest of the application and the bugs most probably lurking there.) The second way to show the value of the test effort is in terms of customer retention with respect to the logon process. I will discuss the CS savings first, because it was very well received by management. Budget and potential or unexpected overruns are something that managers seem to understand vividly, while 'lost customers' or 'potential lost customers' seem to be a bit less 'real' to them.

COST SAVINGS TO CS AS A RESULT OF THE TEST EFFORT

The thing that makes this effort so important for demonstrating risk and its associated costs is that all of these issues were 100 percent reproducible. They were going to occur during *every* enrolment. So *every* customer was going to experience them. Consequently, calculating the value of testing or the risk of not testing in terms of dollars saved was doable, and the results were impressive and defensible.

Developers have long disputed this kind of cost savings statement on the basis that you can't be sure how many users will be affected. If you have to wait until you have collected actual usage stats from production before you can justify an estimate, this usually takes months and by that time, it is hard to get a hearing.

I kept logs of my time spent testing, talking, and reporting. The total came to eight billable hours. For this analysis, I first calculated the cost of the test effort. There were no expenses, so the invoice was for time only:

The cost of my testing: 8 hours x \$125 per hour = \$1,000

The next step was to calculate how much cost was incurred by the company for supporting the end user – me – through this process. I had spent 215 minutes on the phone with the ASP and the trust company customer service accomplishing the enrolment process. Their support costs for call-in customers on a per-minute basis were estimated at \$5 per minute. This figure includes all the costs associated with handling a call, floor space, telecommunications, salaries, benefits, and so on. (*Note:* this estimate is very conservative; larger organizations support significantly higher costs per minute in their call centers.) This gives a typical cost per (computer-literate, web-savvy) customer of:

215 minutes per customer x \$5 per minute = \$1,075 per customer

So, they would probably make back their investment on testing the first time a customer did not have to call customer service to get through the process of setting up their plan. This estimate doesn't take into account all the potential customers who would not become customers because of the number of issues they would have to resolve in order to set up their plan. Nor does it take into account the ones who would simply fail, give up, and go away without calling customer service – more on this topic in a moment.

Next, I asked how many customers they expected to sign up in the first month. (I ignored the cost to fix the bugs and assumed for the sake of estimation that all the bugs I found would be fixed within the first month.) The company had planned a mass mailing to 40,000 targeted prospective customers. They expected a record response to the long-awaited new plan, so estimates went as high as 20 percent response to the mailing, or 8,000 customers for the plan. (A 5-10 percent response would be the normal estimate.) So, I calculated a range from a 5 percent response, or 2,000 customers at the low end, to a 20 percent response, or 8,000 customers at the high end.

For simplicity, I will round off the estimated customer support cost per customer to get through the plan setup stage (calculated in the preceding text) to \$1,000 per customer. The savings in customer support alone, due to testing this one feature, is somewhere between:

\$1,000 x 2,000 customers = \$2,000,000 and

\$1,000 x 8,000 customers = \$8,000,000

Note: conversely and conservatively, the risk of not testing is somewhere between \$2 and \$8 million *in the first month*.

If no testing is done, this range represents the budget that they had better have ready for customer support if they want the product to succeed.

THE USER'S RESPONSE TO FIRST TIME LOGON FAILURE

Logon failure statistics gathered in 1993, 1997, 2000, 2001 and 2002 show the first time user's response to logon failure (Table 1). The 'other' category originally was for users that fell through the cracks and couldn't be accounted for. Starting in 2000 the 'self help' web sites and automated password reissues became common, and have had profound impact on user satisfaction; they are listed in the 'other' category.

The user's response to first time logon failure is possibly the single biggest reason for startup failure in new e-business offerings in the late 1990s. The statistics show that in 1997 over half of all the users who failed in their first attempt to log on to the product or service, left and never returned. The data collected in 2002 suggest that this number has been reduced to about 40%. This seems to be largely due to the improved automated online self-help web sites available now with almost all online services. Typically, the user can ask for a new ID and/or password by filling out a form, and perhaps answering a security question, and the new information is automatically sent to the user via email. The pie graphs in Figure 4 illustrate this change in user behavior between 1997 and 2001.

Year	Network	Leave and Don't Come Back %	Call Customer Support %	Email Customer Support %	Other %
1993	Prodigy	50	45	0	5
1997	Internet IE4	60	5	30	5
2000	Internet e-business	40	15	30	20
2002	Internet e-business	40	15	15	30
2000	Intranet	40	25	25	10
2002	Intranet	20	20	30	30

Table 1: Study Results of the User's Response to First Time Logon Failure Over Time

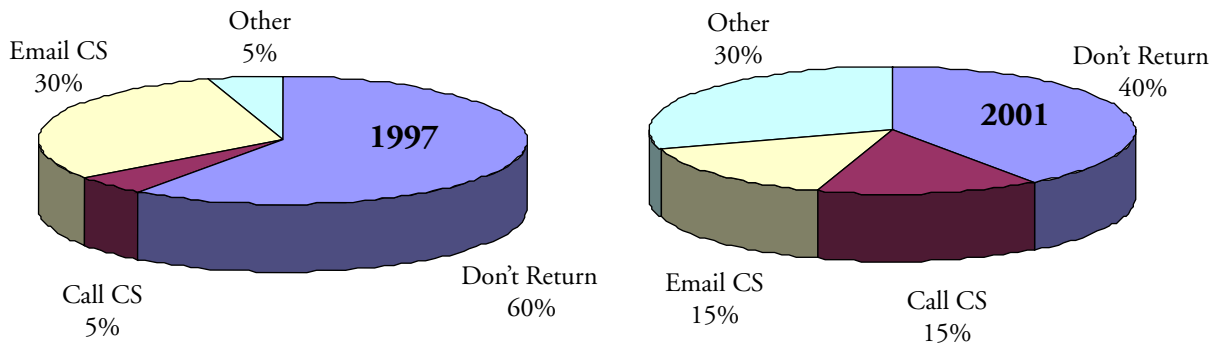


Figure 4: Change in User Behavior Between 1997 and 2001

SOME THOUGHTS ON THIS PROJECT

This was a small application. I walked into it cold and discovered many places where it was broken or disconnected, and I reported them to the maker. There was no planning and no estimation done. I just started exploring. The effort certainly saved the company millions of dollars at the very least. But this is not the best way to conduct a test effort, nor is it the best way to ensure that testing adds value to the project.

The first problem is that in a larger project this would not be possible. While I could manage the complexity of this one function in my head and in my notes, in a larger project there are too many complexities to succeed using this method. Like the hero Theseus in the Maze of the Minotaur, I need a string to help me find my way. This lifeline is a set of best practice risk-based testing methods.

The next point is that this was an ad hoc effort; a one-time deal. Again, ad hoc efforts don't work on a large scale – which brings me to the next problem. The fact is that the work that I do is based on years of disciplined professional practice, and defensible risk-based methods using solid measurement. Management only saw the results of this discipline, not the years of careful exercise that made it possible. They seemed to be left with the impression that anyone with a modicum of experience could produce a similar set of results. This is not the case, and it is this shallow interpretation of the test professional that has led us to the current situation, where developers can say that testers are not necessary and not adding value to the product – and be believed.

The biggest thing wrong with this whole scenario is that it gave management the wrong message, even though it was unintentional. For some reason, they seemed to believe that even though I only tested one-tenth of the application, I had discovered most or all of the bugs and they were now safe to proceed with the deployment. So they were very happy with the savings provided by the test effort, and they deployed the marketing materials that would bring in the mainstream customers.

Without a test effort to warn them of these problems in advance, the company would have discovered these issues in public and in volume. Yet they had not thought it necessary to spend even one percent of this potential loss on a test effort to prevent it. This type of decision is typical of the 'I feel lucky' approach to software development that I first described in 1993 in *Software Testing Methods and Metrics*. The 'I feel lucky' approach is characterized by an entrepreneurial management, with a shallow understanding of the technology, making decisions based on their hunches, rather than real data, or research. This decision is also proof that the 'I feel lucky approach' to software development is still very much in use today.

WHY DOES EXTREME PROGRAMMING WORK?

There are several reasons why XP projects can be deemed 'successful'; not the least of which is the fact that in this brave new Internet world, no one is measuring how much business was lost because of the failures in the software, so any apparent success adds to the bottom line, and few of the failures are counted at all.

If XP succeeds in fielding a viable product it is due to the MFF strategy that goes along with this approach, and the full dress rehearsal given to the software while the first users test it. The successful XP projects that I have seen keep a full contingent of developers at the ready when they roll out the software to a select few users. This gives them the lead time that they need to fix major issues before too many people see the thing. And the users doing the testing get a feeling that they have a very important status – after all, the developers are changing the application to suit them. This generates good will, and gives them a stake in the product.

This contrasts sharply with the plan-driven approach, where the developers who wrote the code have invariably been assigned to their next project as soon as their code is complete, leaving no one who is familiar with the code to fix the bugs. This is especially true and problematic in large integration projects, where integration issues are not discovered

Next column

until well after the code has been turned over. The time required to fix the bug and the cost to fix the bug are both significantly higher than a bug fixed by the original author. Meanwhile, release is delayed, or the application is released and the users can become disenchanted while waiting for fixes.

SUMMARY

After close inspection, I have decided that though the XP tenets are laudable, they will most likely become the mantra for the latest iteration of the ‘I feel lucky’ approach to software development. There are a couple of weaknesses that I believe will cause many XP projects to fail. First, they don’t contain any pledge or commitment to ethical behaviour. Nor are there any balancing activities to ensure the delivery of a functioning product. In short, no fundamental metrics are built in, only subjective judgments. Which brings me to the second weakness ...

There is an implicit assumption in the tenets that the customer knows what they want and that they are willing, able and – most of all – qualified to know what the product should do and what features it should have in order to achieve that goal. This is rarely the case. Users’ expectations are vague and non-specific and, worst of all, these ideas vary from user to user. A design environment dominated by the user’s expectations and the developer’s virtuosity must be strictly managed to prevent scope creep, feature bloat, and other impossible conditions that I call the ‘kid in a candy store’ syndrome. I find it hard to imagine management successfully anticipating and controlling such an environment without the kind of feedback that a professional test effort provides.

XP’s predecessor, the RAD approach, at least pretended to do formal testing before letting the users do the real testing. XP uses a frontal assault on the concept of testing, asserting that the method is so good that no professional testers are necessary – which means that no time or effort is wasted before turning it over to the end user who will begin testing it in earnest, whether they mean to or not.

Even if some developers are adamant that no professional testers are required in an XP effort, there will be those more cautious and mature among the sheep who will be willing to risk a bit on an evaluation. The lead developer in my XP project said this to me:

“We really need testers on the Internet, people have been writing code and sticking it out there for years without testing anything.”

Managers at the ASP and at the trust company didn’t originally believe that a test effort by a professional

tester would be worthwhile. Until my report, the value of such an effort was a mystery to them at best. It is a sad fact that something that can make such a profound difference to the bottom line of a start-up operation as a ‘test effort’ could be so thoroughly discounted and misunderstood. It is my hope that this experience will help other testers to demonstrate the value of their efforts, and that they will pursue every opportunity to demonstrate this value.

REFERENCES

- Agile development methods. See Agile Alliance. “Agile Software Development Manifesto.” February 13, 2001. www.agilemanifesto.org
- Beck, Kent and Martin Fowler. *Planning eXtreme Programming*. Reading, Mass.: Addison-Wesley, 2001.
- Boehm, Barry. “Get Ready for Agile Methods, with Care.” *IEEE Computer*, January 2002.
- Hutcheson, Marnie. *Software Testing Fundamentals*, NY, NY: Wiley, May 2003
- Hutcheson, Marnie *Software Testing Methods and Metrics*, www.ideva.com, 1993-1997
- TestersParadise.com – resources and discussions for testers.

NOTES ON TERMS

The ‘I feel lucky’ approach to software development: term coined by the author in *Software Testing Methods and Metrics*, 1993 and also discussed in *Software Testing Fundamentals*, 2003.

Plan-driven: term coined by Barry Boehm in his article ‘Get Ready for Agile Methods, with Care’ to describe traditional waterfall-style development methods.



A structural engineer by degree, Marnie was recruited into the world of large computer systems by Prodigy Services Company in 1987. She left her job as a field engineer on a high rise building to become the Lead Systems Integrator for Shopping and later Banking and Financial Services at Prodigy. Since the early 1990s she has helped corporations like GTE and Microsoft develop and launch several major Internet technologies. She also creates technical courseware for Microsoft Corporation and travels around the world training the trainers who teach these technologies to the world. Marnie is an internationally published author and speaker in the areas of Software Development and Project Management, Testing and Quality Assurance and Systems Administration. See: www.ideva.com

Next column

Right Product, Right Process

by Daniel Dresner

INTRODUCTION

In this article I am taking up the editor's kind invitation to revisit DTI/NCC's Towards Software Excellence (TSE) resource and look at how its self-assessment technology has been extended to drill down further into the quality characteristics that are so important to software.

Today, we see an apparent distraction from general, grass roots quality issues (represented by ISO 9001, TickIT, and the newly reissued ISO 90003) to more and more specific areas of compliance such as:

- accessibility,
- interoperability,
- security,
- privacy or 'data protection'.

Some of these focus the mind wonderfully when you consider that punishments for non-compliance may vary from a £5,000 fine to a complete loss of business continuity. You can be sure that 'basic' standards aren't going to get a look in. However, when you realise that most of these terms are the nouns which comprise the ISO set of software quality characteristics, this approach

to quality (Aargh! 'ity' again) is far more fundamentalist than developers would be prepared to admit.

WHAT IS TSE?

TSE is a web-based scheme, managed by NCC, that attracted 1,100 registered users in its first two years of operation. The overall objective set for TSE, supported not only by UK government but also other industry bodies, was to enable smaller enterprises in the UK software supplier industry to compare their current approach with best practice. This gives them the knowledge to improve their software processes and hence maintain or improve their competitiveness.

A wealth of good practice has been built up from this experience. Ensuring these practices are available to all, especially smaller enterprises, lies at the heart of the DTI/NCC-established scheme: Towards Software Excellence (TSE). The scheme provides self-assessment, advice, and support over the Internet, aimed at helping smaller software development companies and IT enterprises to understand the capability of their current practices and improve their business processes.

qsigday

Questions Status Route Map Results Report

TOWARDS SOFTWARE EXCELLENCE

Acquisition Preparation

You should carry out this preparatory work so that you can establish the needs and goals of any acquisition - that is, to know why you are buying new, enhancing something you already have or developing something yourself.

Q1 Do you assess a customer requirement for your ability to provide the solution from within your own capability and/or buy an external solution?

Always Usually Sometimes Rarely Never N/A

Q2 Do you produce a description of the customer's requirements (a requirement specification)?

Always Usually Sometimes Rarely Never N/A

Q3 Do you use a process to agree with the customer that the requirement specification is accurate?

Always Usually Sometimes Rarely Never N/A

Q4 Does the customer sign the requirement specification to show their agreement?

Always Usually Sometimes Rarely Never N/A

Q5 Do you use a pre-defined process to agree whether the 'acquisition' will be a purchase or in-house development / enhancement?

Figure 1: Simple On-line, Self-assessment in the ISO 15504 Mould

The scheme operates by providing its users with three key components:

- a route map through best practice, based on international software process standards,
 - a self-check tool by which an enterprise can evaluate its current software practice and subsequently measure its progress towards improved processes (and ultimately achieve software excellence),
 - benchmarks and self-certification that suppliers may use to prove to customers and partners their competence, effort and attention to quality.
- TSE is designed to:
- cut development times and costs,
 - improve market delivery,
 - ensure smoother customer relationships,
 - compare performance with equivalent developers or sectors,
 - prevent software-related failures.

The impetus for TSE came from the DTI, which recognised how the software industry had learnt much from developing the programs that lie at the heart of our everyday lives – from controlling traffic lights to running elevators; from powering microwave ovens to flying jumbo jets. When it goes wrong, lives may be lost, businesses can fail. TickIT had been established to make sure that these lessons can be applied in the business environment, but smaller organisations needed a leg up to reach the first rung of the quality ladder. DTI put up half the funds for the scheme's start-up, since it links in with government objectives to ensure the UK is at the leading edge of the new economy. The rest was provided by The National Computing Centre.

FAVOURITE THINGS

In 2003, The National Computing Centre published its framework for managing ICT risk in the guise of

'Desert Island Standards' (Guideline 275). This report asks you to imagine you're suddenly called to manage the information technology on a desert island and you're only allowed to take eight standards with you to mitigate risk. Which standards would you take? Just as in the BBC set-up you can have the equivalent of The Bible and the Complete Works of Shakespeare, in this fantasy you get:

- ISO 9001 for Quality Management and
 - BS 7799 (or the international version ISO 17799) for Information Security Management.
- The other standards selected were:
- ISO 15288: Information Technology – Life Cycle Management – System Life Cycle Processes which creates the management structure for projects from soup to nuts,
 - ISO 9126 Software engineering – Product quality which creates a common language for specifying software,
 - BS 15000 IT service management which manages the risk to suppliers and users of information system services,
 - ISO 15504 Information technology – Software process assessment which creates the environment for benchmarking and ensuring that risk management initiatives contribute to continuous improvement,
 - The Data Protection Act 1998 which manages the important social risk of privacy that can be compromised so quickly for so many by so few with information technology,
 - STARTS Software Techniques for Reliable, Trusted Systems which reduces the risks derived from adversarial relationships between suppliers of information systems and their customers,
 - ISO 18019 Guidelines for the design and preparation of user documentation for application software which manages the fundamental risk of systems

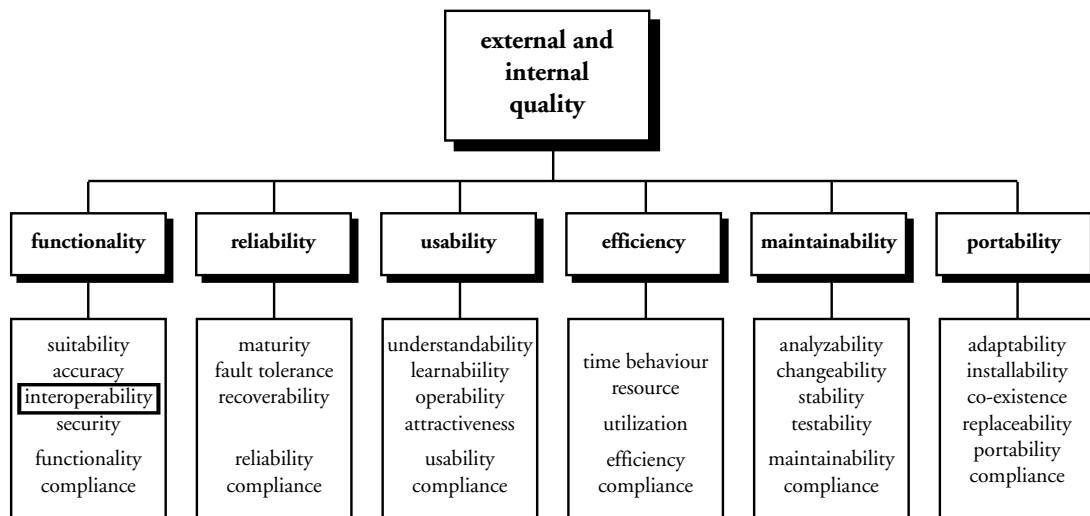


Figure 2: Extract from ISO/IEC 9126

lacking the correct instructions for use,

- e-GIF (the e-Government Interoperability Framework) which reduces the high risk of systems failing to work together.

On the premise that standards come in two flavours – process and product – this list contains the basic ingredients of ISO/IEC TR 15504, where you are asking, ‘Are we doing it right?’ and ISO/IEC 9126 where you are asking, ‘Are we doing the right thing?’.

The process model of ISO/IEC TR 15504 is reflected in the on-line self-assessment of TSE. One of the characteristics of software quality set out in ISO/IEC 9126 is the interoperability within and between systems.

INTEROPERABILITY

The Cabinet Office e-Government Unit (formerly known as the Office of the e-Envoy) has provided a world-class service by designing the e-Government Interoperability Framework (e-GIF). Don’t be distracted by the word ‘government’, it is actually a generic model for interoperability.

It’s a bit of a chutzpah including this in the list of eight standards to take to the desert island. It’s a bit like asking your fairy godmother to make the third wish a thousand more wishes. It’s not really a standard but an umbrella covering many standards that comprise the e-GIF. You can extrapolate a lot from this framework for interoperability ...

What is the Objective of the Standard?

Adherence to electronic-Government Interoperability Framework (e-GIF) specifications and policies is mandated for new systems and legacy systems involved with electronic service delivery targets and for the exchange of information between government systems and the interactions between them. So if two parts of an IT jigsaw comply with e-GIF, they should be able to work together. It’s a great vision – obviously some of the STARTS leadership vision at work here – and the e-GIF framework makes it achievable. The e-Government Unit, which has defined the e-GIF, has set the goal as an environment of efficient transitions and improvements to e-Government for:

- UK Government and citizens,
- UK Government and businesses (worldwide),
- UK Government organizations,
- UK Government and other governments (UK/EC, UK/US and so on).

e-GIF is drawing international attention already as other governments and CEN see its potential and look to adopt what is really a generic technical framework for interoperability to connect:

- an organization to its local customers,
- an organization to its International customers,
- an organization and its supply chain,
- collaboration between supply chains.

Next column

Why Is It So Good?

It recognises the complexity of IT and the need to manage the risk in this complexity (remember ISO 15288). e-GIF sets boundaries to make systems work together by identifying where connections must be made, rather than setting out to make all connections work – some of them may never be used. The framework of standards has built in its own robust, self-sustaining process to ensure that it is managed as a long-term, ongoing initiative as technologies change and new standards are distilled from the quagmire of innovation and experience. This process, including the roles and responsibilities of key stakeholders, committees, management and working groups, is outlined in the e-GIF documentation. This change management is designed to engage and serve the stakeholder community in a dynamic way and to bring in innovations from industry on a global basis.

How Do You Use It?

It’s a ‘pick and mix’ catalogue for you to use to select the appropriate components for what you want to achieve. For example, at the highest level complying with the e-GIF means:

- providing a **browser** interface for access,
- using **XML** as the primary means for data integration,
- using Internet and **World Wide Web standards**,
- using **metadata** for content management, and
- can any **component** or product used within an interface be **replaced by** another of a similar specification and the functionality of the system still be maintained?

What’s In It?

The main thrust of the framework is to adopt the Internet and World Wide Web specifications for all government systems. This takes advantage of the greatest outsourced networking opportunity ever – the Internet. e-GIF systems include their interfaces, hence the strategic decision to adopt XML and XSL as the core standards for data integration and management of presentational data.

The framework also sets out policies for establishing and implementing metadata across the public sector. The application of the e-Government Metadata Standard, for example, will help citizens find government information and resources more easily.

Stipulating policies and specifications in themselves is not enough. Successful implementation requires the provision of support, best practice guidance, toolkits and centrally agreed schemas. This is a subject close to the heart of NCC and so, building on the technology that delivered the practical testing and advice of TSE, NCC developed a self-assessment resource for e-GIF – interoperability – compliance.

This is coming full circle back to TSE with the



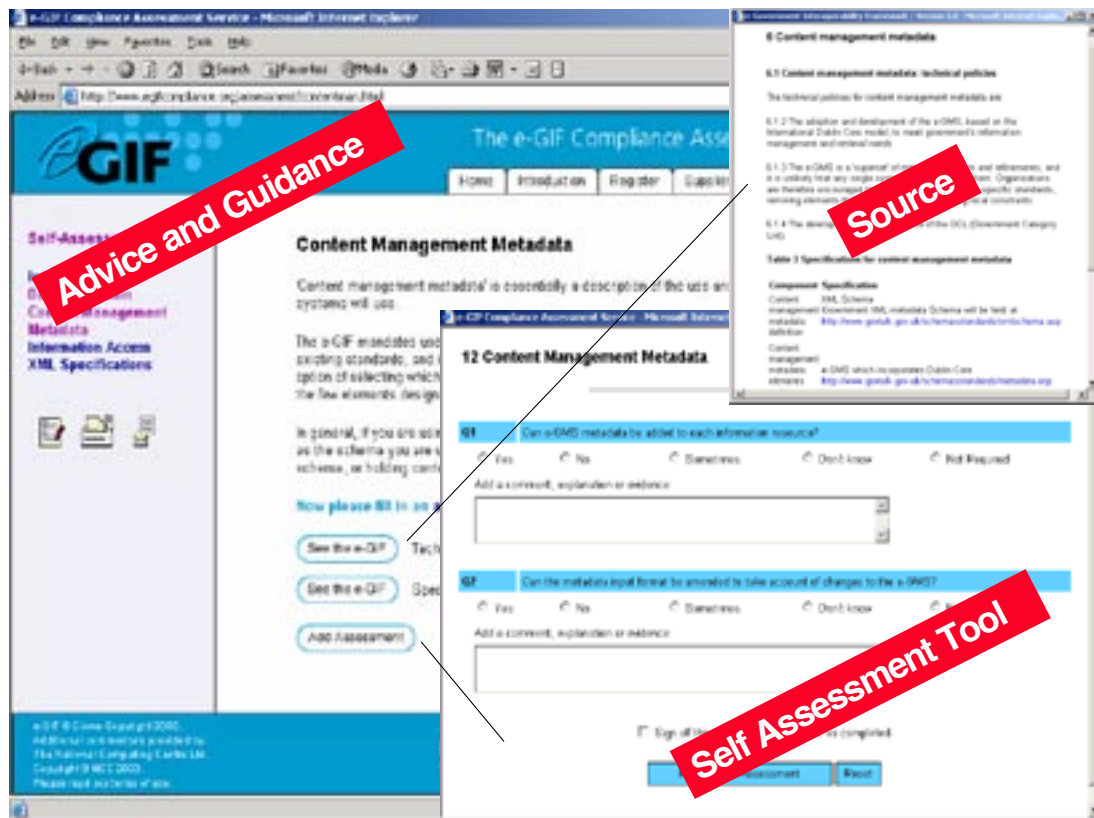


Figure 3: Interoperability Assessment Driven By e-government

creation of the e-GIF accreditation body, which will not only certificate practitioners with relevant skills in interoperability, but will also accredit organisations who demonstrate that their processes meet a well-defined standard.

WHERE TSE IS SUCCESSFUL

Feedback from TSE users suggests that it makes a good introduction to breaking up software procurement and development into a life cycle, and providing guidance for developing customized templates. A recent 'convert' dived in the route map to glean guidance on managing the tendering process for a new system.

Significantly, TSE is free from the pressures of certification, and complements existing schemes such as ISO 9001/TickIT and the Capability Maturity Model (CMM) – although it is hoped that it will encourage SMEs to take up such schemes when they feel the time is right and they have the resources available.

WHERE MIGHT TSE GO?

When you look at the quality model in ISO/IEC 9126, you can see the range of characteristics that, once defined, need to be inspected and ultimately tested. www.e-GIFcompliance.org has interoperability well covered but what of the others? Perhaps this is the future, not only of TSE but of the approach to software quality.

Editor's note: for those who wish to view the source go to www.govtalk.gov.uk

Now it's up to the users. There's a big stick for interoperable, e-GIF compliance by 2005. This is generating the subscription funds to develop the interoperability test set. The test schedule has recently been the target of a major revision from a few generic questions to a complete set of self-assessment questions for e-GIF version 6.0. These are not only a step up in terms of keeping up to date with e-GIF, they are also a significant change in the level of detail required. This will make assessments more accurate, whilst maintaining the flexibility of the original assessment service which called on you to check e-GIF compliance at a very high level. As with TSE, a summary report is delivered in the context of your projects and this can be used as a certificate of compliance.

TSE is free to NCC members and just £35 + VAT per seat for everyone else. See <http://www.software-excellence.org.uk>. Carrots and sticks this way please.



Daniel Dresner is Standards Manager at The National Computing Centre (daniel.dresner@ncc.co.uk)

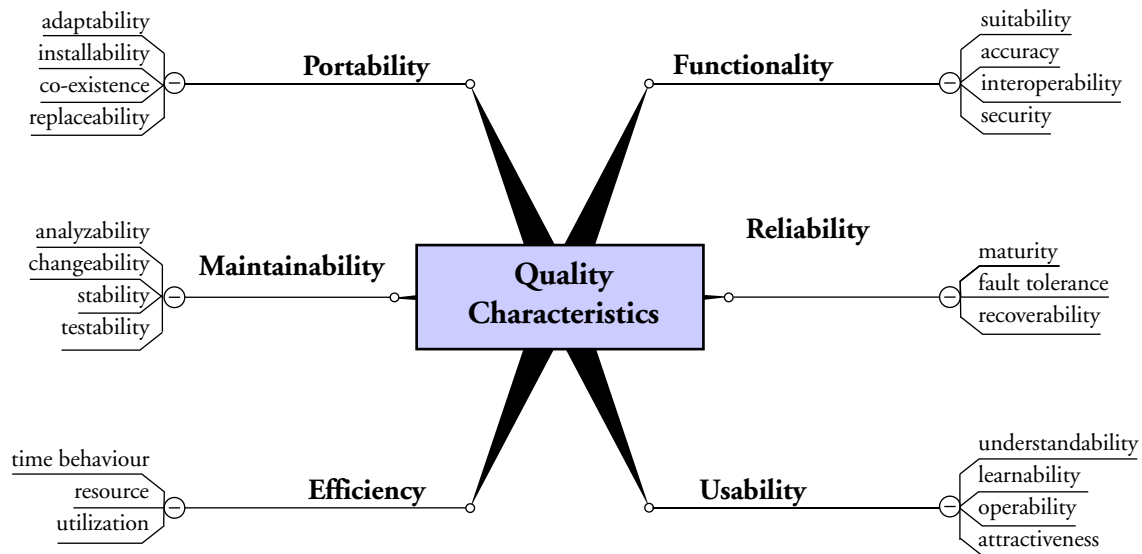


Figure 4: ISO/IEC 9126-1 Software Engineering – Product quality – Part 1: Quality model for external and internal quality

A Report on the BCS Quality SIG's SQM 2004 Conference

By: Margaret Ross

The British Computer Society's Quality Special Interest Group held its 12th annual International Conference on Software Quality Management, SQM 2004, on 5/6th April at Canterbury Christ Church University College.

The conference was attended by delegates from industry and academia, with speakers from India, Australia, Canada, Spain, USA, Finland and Greece as well as the UK.

Keynote

The highlight of the first day was the keynote speech by Prabhu Sinha, Senior Vice President (Worldwide) of Satyam Computer Services on Global Trends in Software Process Improvement. This was very well received and was followed by an interesting discussion.

First Session – Outsourcing

This began with Clifford Liles of Logica CMG discussing how the risks of outsourcing could be quantified. Valeria Edgar-Nevill presented a case study of the use of outsourcing to transform an organization's business processes.

Second Session – Standards

This session was opened by Robert Martin of the Mitre Corporation, USA, with a presentation on two international initiatives relating to vulnerabilities in commercial and open source software. The Common Vulnerabilities and Exposures (CVE) initiative is creating an organizing mechanism to make finding and fixing vulnerabilities faster and more efficient. The Open Vulnerability Assessment Language (OVAL) provides a means for standardized vulnerability assessment.

Josephine Wapakabulo of the LSC Group presented a model of the factors that influence the adoption and diffusion of standards.

Antonia Mas and Esperanza Amengual of the Universitat de les Illes Balears, Spain discussed the implementation of a quality management system developed for use by small/medium-sized software houses.

Sylvie Trudel, of the Software Test Centre, CRIM, in Canada outlined the design of a software quality evaluation method combining CMMI and ISO 14598.

Third Session – Product Quality

Robert Martin of the Mitre Corporation, USA, discussed the Industrial Applicability of ISO Standards on Software Product Quality.

The paper of Elankayer Sithirasanen and Jennifer Gasston of Griffith University, Australia demonstrated that the current process models, such as CMMI and ISO 15504, did not support the ‘engineering-in’ of quality into a software product during its development.

The paper of Witold Suryn, Alain Abran, Pierre Bourque and Claude Laporte of the École de Technologie Supérieure, Canada, described an integrated life cycle quality model, which combines the high-level quality view of the TL 9000 Quality System Requirements for Telecommunication Products (including software) and the more detailed view from ISO 9126. The process of building quality into software during development was also discussed.

Fourth Session – Process Improvement

Margaret Ross discussed the effect that the introduction of a Project Management Office has on a Quality Management System, based on the experience of a large multi-national organization.

Dili Ojukwu of Middlesex University presented a case study of small- and medium-sized enterprises in Nigeria, and described how quality improvement of their goods and services was achieved using the concept of Integrated Business and Information Solutions.

Birinder Sandhwalia described how knowledge and learning could be incorporated into the software development process, using a Dynamic Feedback Model.

Peter Bennetts of the University of Gloucestershire discussed the use of Soft Systems approaches in Software Process Improvement. He argued that a purely technical approach to SPI was not sufficient, and that using soft systems approaches would address effectively the human and organisational issues.

Fifth Session – Re-engineering

Elli Georgiadou of Middlesex University described a case study of the migration of a structured system design to UML models. This enables legacy systems developed from a structured design to be converted to object oriented systems without losing data or functionality. Proposed guidelines for the re-engineering process were also presented.

Henri Basson of the Laboratoire Informatique du Littoral, France, presented a model based on graphics and XML that deals with both high-level and low-level software artefacts in a uniform way. He described how the model has been used to implement tools for change impact propagation, and for recovering the architecture of a design from source code.

Next column

Sixth Sessions – Metrics

John Moses of the University of Sunderland discussed the problem of assessing variation in development effort consistency when some of the data are missing. He showed how Bayesian Inference could be used to deal with this situation.

Qin Liu of the University of Northumbria described a statistical analysis of a historical data set that is widely used as a basis for the estimation of software development effort and duration.

Boris Colgan of London Metropolitan University presented a generalised structural model of object-oriented programs, and discussed its use in defining object oriented software metrics.

Seventh Session – Quality Issues in e-Commerce

Tom Jackson of Loughborough University discussed the use of Instant Messaging as a communication tool for business. He compared Instant Messaging with other forms of communication, including its associated interrupt recovery time.

Lisa Liu of Glasgow Caledonian University discussed the identification of risk factors in software projects in an e-business environment in the Financial Services sector. Based on a detailed survey of sixteen software projects, she identified the principal risks, and discussed the factors that might reduce these risks.

Anthony Burgess discussed the effectiveness of training in reducing the occurrence of email defects.

Cheong Ching Dang presented a pragmatic model for managing knowledge in online communities. She presented a model that allows organizations to manage their knowledge in a well-defined structure, and to capture metrics to gauge the effectiveness of knowledge capture and sharing.

Tonderai Maswera presented the results of a survey on the accessibility and usability of the websites of tourist organizations in four African countries.

PRIZE-GIVING

Pfizer Limited sponsored a prize for the best paper at the conference. Two papers were identified as equal ‘best’, and so both were presented with an engraved glass vase. These were:

- A Generalized Structural Model of Object-Oriented Programs for software metrics Definition: [Boris Cogan (London Metropolitan University, UK), Elena Shalfeeva (Institute for Automation and Control Processes, Far Eastern Branch of the Russian Academy of Sciences, Russia)]
- The Design of the Software Quality Evaluation Method combining CMMI and ISO/IEC 14598: The context of a small company. [Sylvie Trudel (CRIM, Canada), Jean-Marc Lavoie (Consultations Jean-Marc Lavoie Inc. Canada), Marie-Claude Paré (Motorola Canada Software Center), Witold Suryn (École de Technologie Supérieure, Montréal, Canada)].



Copies of the proceedings for SQM 2004 are available at £30 including post and packing (additional postage cost to be added for overseas purchases). These are available from Margaret Ross (margaret.ross@solent.ac.uk) on receipt of a cheque payable to 'BCS Quality SIG' or by credit card authorization.

SQM/INSPIRE 2005

Abstracts are now being accepted for both the two-page Management Summaries and for the full 12 page

papers for the 13th Software Quality Management conference, to be held in Cheltenham from Monday 21 to Wednesday 23 March, 2005. Abstracts, of approximately 500 words, should be sent to Margaret Ross.

The conference web page, call for papers, proposed subject areas and submission details can be found on: go.to/sqm or: <http://online.glos.ac.uk/sqm>

See the website for full programme details, addresses and other administrative information.

Rumblings of a TickIT Trainer

By Neil Martin

TickIT has now been around for the best part of 14 years. In the early days I was working on software product assessment for BSI, an organization that was not particularly loved by those responsible for setting up TickIT. Since then, I have seen many sides of TickIT and related activities. I did my share as a quality manager obtaining NAMAS accreditation for the Software Test Laboratory at BSI; later I set up my own company obtaining TickIT Registration. Since then I have helped with certification bodies seeking TickIT accreditation and end companies seeking TickIT registration.

My main involvement with the TickIT Scheme at the moment is delivering TickIT Lead Auditor courses for two providers in the UK, namely LRQA and PERA Neville Clarke (in strictly alphabetic order you understand). Over this period I have experienced the highs and lows of the TickIT Scheme, and while delivering the Lead TickIT Auditor course I have heard many positive and negative things said about TickIT. This article is a summary of these issues, intended to generate discussion then – with a bit of luck – some improvement in places.

THE TICKIT SCHEME

Prior to my setting off on a number of different issues, it is probably worthwhile revisiting some of the history of the TickIT Scheme; for many readers this will be a recap but for others it may not be as familiar.

Background to The TickIT Scheme ...

It was back in 1990 that the UK Department of Trade and Industry launched the TickIT initiative. In the context of ISO 9001 certification, TickIT is an accreditation scheme. This means that it is a process for attesting to the competence of a certification body to audit organisations that are involved in IT

Next column

and software engineering. It is not, as many often still believe, a standard against which organisations that are active in areas of IT and software engineering have to comply or – for that matter – are audited against. The document ISO 9000-3:1997 (now replaced by ISO 90003:2004) is a guidance document and does not add to or subtract from the requirements of the standard ISO 9001:2000, which applies equally to IT and non-IT organisations. Despite what some people may think, this does make sense, as ISO 9001:2000 is a set of standard requirements on a quality management system, most of which is pretty much the same regardless of the type of organisation and – I hasten to add – pretty much common sense for most organisations. In addition to the accreditation aspects of TickIT, the DTI used it as an opportunity to promote best practice in software development, mainly through the distribution of the TickIT Guide. This Guide gives guidance to purchasers of IT systems, suppliers of IT systems and auditors of quality management systems.

TickIT Accreditation

This process involves a certification body demonstrating established processes for certification, and that the auditing staff and certification officers that it employs have a basic competence through the use of TickIT auditors. The process is similar to ISO 9001 assessment but is in fact against EN 45012 (EN means Euro Norm); 45012 is General criteria for certification bodies operating quality system certification, rather than ISO 9001. The audits (in the case of the UK) are performed by the United Kingdom Accreditation Service (UKAS) and in the case of Sweden by the Swedish Board for Accreditation and Conformity Assessment (SWEDAC).

In short, accreditation of certification bodies is intended as a safeguard for companies purchasing certification services.





Figure 1: TickIT Accreditation/Certification

TickIT-Accredited Certification

IT organisations that are registered under the TickIT Scheme are audited against ISO 9001:2000 by a TickIT-accredited certification body. There are no additional or lesser requirements on these organisations than any other type of organisation that may seek ISO 9001 certification. One of the key issues for the TickIT Scheme is the competence of auditors.

Now that we have reminded ourselves of the basis of TickIT, let me visit some of the issues that arise during the delivery of TickIT training courses:

... and Back to the Article

WHAT IS TICKIT?

I am now used to this fact, but it still troubles me, that well over half the delegates who turn up on a Lead TickIT Auditor course do not know at the time of booking what TickIT is. (I think this indicates that the message of exactly what TickIT is, is confusing and work needs to be done on explaining it.) The majority of people have a simple understanding which is basically that TickIT = software quality assurance. Why is this a problem? Well, the TickIT Lead Auditor course is intended to train third-party TickIT auditors. Strangely, trainee certification auditors are not something we see many of on the TickIT Lead Auditor courses!

Registered TickIT Auditors

Both the TickIT office and IRCA have complained at various times about the lack of funds to promote and support the TickIT Scheme, yet despite this the TickIT Scheme still only recognizes third party auditing and fails to offer a registration scheme for recognition of software auditors who are active in industry. To emphasize this point, about 99 out of every 100 delegates that attend the TickIT Lead Auditor courses are people working in industry; many, but not all, are internal auditors. It should come as no surprise, then, that the number of people registering as TickIT Auditors is small and diminishing; many from industry view the whole thing as a cartel and simply do not

Next column

understand why there is no recognition of competence of auditors working in industry. (*Note:* IRCA do have a curriculum for an Internal TickIT Auditor course which is run by a number of training providers, but IRCA does not register the auditors passing these courses under the TickIT brand. I do not understand why they don't see this as a commercial opportunity)

Certification Bodies and TickIT

This may come as a surprise, but many certification bodies actively dislike TickIT. There are no doubt many reasons for this, but one of the primary reasons seems to be that having to select auditors based on skill set rather than post code is inconvenient for their planning arrangements! It would be far simpler to send the nearest auditor rather than the most appropriately trained auditor.

Note: The issue of a TickIT-accredited certificate without the presence of a TickIT Auditor would be fraudulent behaviour by the certification bodies. Now, you can accuse me of bias or vested interest (since I am a TickIT Lead Auditor) but I believe that TickIT registration specifically, and registration in general, would benefit from a process that included the publishing of the auditor's name alongside a company's initial registration details – at least that way we would all know publicly that trained auditors had been involved.

Improvement of Software Development/Software Quality

If one recalls that one of the motivations for setting up TickIT was to encourage improvement in the overall quality of software being developed by UK plc, it would be helpful if data demonstrating the improvements made since the introduction of TickIT were readily available on the TickIT website or other locations.

Software Metrics

The TickIT office publishes a document called *Getting the Measure of TickIT* that gives guidance on various ISO standards related to software measurement, but this is still an area that many organisations struggle to deal with in a satisfactory manner. ISO 9001:2000 has 'upped the ante' on metrics and requires both product and process measures. However, from the very un-scientific surveys of delegates attending courses it does seem that many TickIT certification bodies seem to have taken the view that all that needs to be done from their perspective is to raise a nonconformity against measurement. I am not sure how they think this will help, but I suspect there is an attempt to wear a Teflon coat.

The other area of software metrics that is a problem, and it may disappoint many if I say so, is that there seems little evidence as yet that ISO standards such as ISO 12207:1995 'Software Life Cycle Processes' and ISO 15939:2002 'Software Measurement

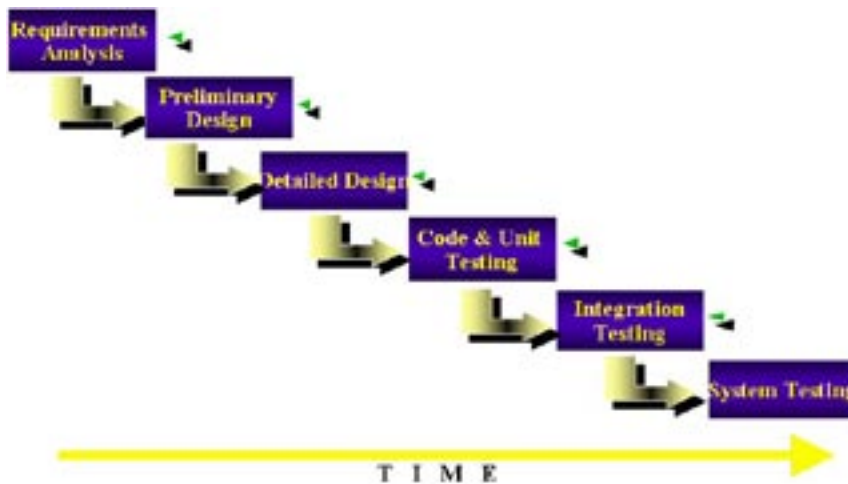


Figure 2: Waterfall Life Cycle

Process' have made any impact. The ISO committee JTC 1/SC7, as readers of this magazine will know, does an unbelievable amount of work on software metrics, yet there seems little evidence (yes, my unscientific measures again) that this is converting into useful, value adding, processes for business. What I believe is needed is more examples of successful approaches to measurement of software processes, rather than more standards or guidance.

Changes to Software Life Cycles

This is an area similar to software metrics that has changed very little over the last 10 or more years. Although documents and guidance such as ISO 12207 and its friends and family exist, many organisations still use a variant of the waterfall – see Figure 2 – (Winston Royce 1970), a development model described as unlikely to succeed in any real implementation.

(see <http://c2.com/cgi/wiki?WaterFall>).

This is not to say that many organisations are not using incremental models, or other models, but there are still a large number of organisations that are unaware of anything better than a model discredited in 1970.

The area of software life cycles is curious; in the last ten years there seems to have been almost stagnation in this area. I know that Rational Corp (see <http://www-306.ibm.com/software/rational/offerings/lifecycle.html>) through their Unified Process and Extreme Programming (<http://www.extremeprogramming.org/>) have at least moved things on, but as far as I know we do not have a non-proprietary off-the-shelf life cycle that embraces the 'best of breed' for software development. There also seems to be very little evidence of computer science graduates coming out of further education with a decent understanding of what it takes to produce a quality software product, rather than just being able to churn out code.

Next column

Relationship With CMM

Some managers view CMM or ISO superior to the other, normally for ill-founded reasons. I believe most quality practitioners would consider CMM assessment and TickIT-accredited ISO 9001 certification to be complementary rather than competitive. However, this does not stop ill-informed managers arguing fruitlessly ...

TickIT Auditor Training

The current training criteria for TickIT courses (IRCA 2212 March 2001) concentrate solely on auditing and do not deal with knowledge of software and software engineering. While this is understandable from an IRCA perspective, since they are in the business of auditor certification not software,

many attendees at the TickIT courses are surprised at the lack of software-related content both in the courses and – more specifically – the exam. I believe the TickIT courses would benefit from some minimal software engineering content as a precursor to the auditor course for those that lack the software background. In theory, of course, to become a TickIT Auditor there is a prerequisite of a track record in software. However, if we simply take the position that auditing software is only open to those that already have experience in software, we immediately alienate a large number of people in the industry. These are already qualified and experienced auditors who wish to learn enough about software development practices to a level sufficient to allow them to operate effectively as auditors in an IT environment. I would argue that there is room for two tracks of auditor; those without the track record in IT should be able to demonstrate competence through a traditional exam.

Let us face it, though, the body of knowledge of how to develop software has increased in a substantial manner since the start of the TickIT Scheme; the exams although now addressing ISO 9001:2000 rather than ISO 9001:1994, are basically unchanged.

TickIT on the Cheap

This leads me to the curious developments in the area of TickIT accreditation of certification bodies at the start of 2000. At this time, UKAS published a document (www.ukas.com) called CISI.pdf and has the title *Use of TickIT as the Benchmark for Software*. The first few paragraphs of this document read as follows:

"In November 1997, the Policy Advisory Committee of the United Kingdom Accreditation Service advised that it was inconsistent with international obligations to maintain TickIT as a mandatory scheme for the certification of organisations involved in software development.

As a consequence, it was resolved that certification bodies would be free to offer alternative services as desired by their clients. However, recognising the concern from main buyers that such a change in policy must not undermine TickIT through the possible availability of less competent certification, it was also resolved that UKAS would use the TickIT Scheme as the benchmark for all alternative accreditations of software certification.

Following consultation with TickIT and the industry, this paper was produced. It identifies those elements of the TickIT Scheme which will be benchmarked by UKAS to ensure that alternative certification services are being provided to a comparable standard."

What does all of this mean for those that might be reading this article without wearing an accreditation and certification anorak? Well, in short, it means that UK-accredited certification bodies are free to offer services equivalent to TickIT, but do not fall under TickIT. I must confess when you read CISI from UKAS, the old joke about 'it walks like a duck, it quacks like a duck, it's a duck' quickly comes to mind. When you read what UKAS says, it sounds like you can run a certification scheme for software, it doesn't have to be TickIT – but it has to be the equivalent to TickIT. I would sure like to hear from people that have obtained registration under such an arrangement. The only case that I know of is that one UK-accredited certification body offers a service that is cheaper if it is non-TickIT. Given that everything is meant to be equivalent, I would be interested to know where the savings are made. It does seem, at least to me, that UKAS have got themselves into a position where they are slowly

undermining the whole TickIT Scheme. It looks like they currently will accredit certification bodies that are using non-TickIT auditors – and probably non-IRCA registered as well. The route that should have been taken, of course, was to uphold the general principle of specifying definitively the training requirements for all types of third party auditors, and then pressing hard on the certification bodies to drive standards upwards rather than allow this gradual dilution process.

Finally, I have made numerous references to my unscientific methods for collecting data. If you would like to help me correct this, I have three surveys related to TickIT running on my website at www.thetestplace.com/TickIT/TickIT.asp

The three surveys are:

Survey 1: TickIT Certification and TickIT Auditors

Survey 2: Implementing a TickIT QMS

Survey 3: The Success and Impact of TickIT Certification

I would appreciate it if readers could find the time to complete one or more of the surveys, and once we get a decent sample size I will produce an article based on the findings for TickIT International. If you wish to comment on what I have said above you can send an email to neilm@thetestplace.com



Neil Martin is a freelance trainer, teaching C, C++, Java and The Lead TickIT Auditor Course. He was the former convenor of the BSI Standards panel IST/-/14 for the C Language and he is an IRCA Registered Lead TickIT Auditor and member of the ACCU (www.accu.org). His company is called TFJ Ltd whose web site is at www.thetestplace.com. He was formerly a director of Plum Hall Europe Ltd and a manager of BSI Quality Assurance Software Products Services (responsible for conformance testing programme including the European C Compiler Validation Service).

Software Measurement

Do your capabilities match ISO 9001:2000 requirements?

Measurement programme consultancy, support, set-up

Visit

www.qasoft.demon.co.uk

DIY WORKBOOKS save time and money

ISO 9001 TickIT
ISO 9001 RAD
ISO 9001 SSADM
ISO 9001 PRINCE
ISO 9001 Service Delivery
Listed on the Internet

http://www.nvo.com/management_systems-author

E. Sutherland, IEE Ind Aff
Trog Associates Ltd
P.O. Box 243
UK – South Croydon/Surrey, CR2 6NZ.
Tel +44 (0) 208 786 7094
Fax +44 (0)208 686 3580
Email trog@dial.pipex.com

ISO 9000/BS7799

Guaranteed Results!

- 100% clients registered first time
- Consultant/Trainer since 1990
- Lead TickIT Auditor since 1992
- TEC & Business Link funded projects
- Practical advice and training

Consultancy Training Pre-Assessments

Bal Matu BSc(Hons) C.Eng.MIIEE FIQA MAQMC

Lead TickIT Auditor

Tel/Fax: 44 (0)1928 723701

bal@bsmqdemon.co.uk



IT 3Q04