18

June 2012

testing experience

The Magazine for Professional Testers

Test Center of Excellence

How can it be set up?

printed in Germany

print version 8,00 €

free digital version

www.testingexperience.com

ISSN 1866-5705

Certified Agile Tester

Pragmatic, Soft Skills Focused, Industry Supported

CAT is no ordinary certification, but a professional journey into the world of Agile. As with any voyage you have to take the first step. You may have some experience with Agile from your current or previous employment or you may be venturing out into the unknown. Either way CAT has been specifically designed to partner and guide you through all aspects of your tour.

The focus of the course is to look at how you the tester can make a valuable contribution to these activities even if they are not currently your core abilities. This course assumes that you already know how to be a tester, understand the fundamental testing techniques and testing practices, leading you to transition into an Agile team.

The certification does not simply promote absorption of the theory through academic mediums but encourages you to experiment, in the safe environment of the classroom, through the extensive discussion forums and daily practicals. Over 50% of the initial course is based around practical application of the techniques and methods that you learn, focused on building the skills you already have as a tester. This then prepares you, on returning to your employer, to be Agile.

The transition into a Professional Agile Tester team member culminates with on the job assessments, demonstrated abilities in Agile expertise through such forums as presentations at conferences or Special Interest groups and interviews.

Did this CATch your eye? If so, please contact us for more details!

Book your training with Díaz & Hilterscheid!

Open seminars:

23.–27.07.12 in Munich/Karlsruhe, Germany
27.–31.08.12 in Frankfurt, Germany
24.–28.09.12 in Stuttgart, Germany
29.10.–02.11.12 in Berlin, Germany
03.–07.12.12 in Cologne/Düsseldorf, Germany

(German tutor and German exam)

11.–15.06.12 in Stockholm, Sweden
02.–06.07.12 in Orlando, USA
03.–07.09.12 in Helsinki, Finland
15.–19.10.12 in Oslo, Norway
26.–30.11.12 in Amsterdam, Netherlands

Díaz Hilterscheid

Díaz & Hilterscheid GmbH / Kurfürstendamm 179 / 10707 Berlin / Germany
Tel: +49 30 747628-0 / Fax: +49 30 747628-99
www.diazhilterscheid.de training@diazhilterscheid.de

Editorial

Dear readers!

Work with our editorial board started with this issue. It is really a pleasure to work with them and to see how close they are marking the articles. It was and still is a good idea to have all of them.

What you have in your hand or on your screen is the work of many authors, media designers, lectors and the editorial board. Thanks to all of them for another great issue.

This issue is about how to set up test centers of excellence. There are many articles telling you how this has been done and in which traps some of the guys stepped into along the way.
The next issue will about mobile apps testing. I'm really looking forward to seeing your articles. I have been testing some mobile apps myself for our conference, and also apps for friends of mine like TestGloss (an app for the ISTQB® Glossary), ePaperExam (ISTQB®exam app), etc. It was fun for me to be "working" again! ;-) Bug hunting is such a nice pastime.

I'm writing this in London, facing the Testing & Finance. Things are looking good. We have a full house with exhibitors, speakers and attendees. The venue is great, even if the breakfast is tooooooooo expensive!

We think that Testing & Development, due to more and more agility in the methods and techniques, will come steadily closer. We will have the Agile Testing Days in November 2012 and have just started the run for the call for papers for the Agile Dev Practices conference. The first papers have already come in. The community is quite well connected and is based on a will to learn more with a view to improvement. I like that very much. Have a look at www.agiledevpractices.com

Last but not least, I thank all the sponsors for supporting the magazine.

Enjoy reading!

Best regards

José

Ranorex®

Automate Testing of any Windows, Web & Mobile App

- ✓ Use connectors for data-driven tests
- ✓ Build robust test automation frameworks
- ✓ Generate EXEs for pure flexibility
- ✓ Write custom code in C# or VB.NET

Record and Edit Reliable Test Actions

Manage and Execute Your Automated Tests

Reproduce Bugs and Maintain Your Tests

Award-winning test automation tools, which allow testing of many different application types, including: Web browsers (IE, FF, Chrome and Safari), WPF, Flash/Flex, Silverlight, Qt, SAP, .NET, 3rd Party Controls and Java.

Download your 30-day Trial at **www.ranorex.com**

testing experience

Contents

Editorial Board

Graham Bath

Graham's experience in testing spans more than 30 years and has covered a wide range of domains and technologies. As a test manager he has been responsible for the testing of mission-critical systems in space-flight, telecommunications and police incident-control. Graham has designed tests to the highest levels of rigour within real-time aerospace systems.

As a principal consultant for T-Systems, the systems division of Germany's leading telecommunications company Deutsche Telekom, he has mastered the Quality Improvement Programs of major companies, primarily in the financial and government sectors. In his current position Graham is responsible for the test consulting group. He has built up a training program to include all levels of the ISTQB certification scheme, including the new Expert Level, for the company's large staff of testing professionals.

Graham is the chairman of the ISTQB Expert Level Working Party and co-author of both Advanced Level and Expert Level syllabi.

Gualtiero Bazzana

has been working in the IT and testing domain for the last 20 years; he is author of more than 50 publications on the topics of sw quality and testing; he currently holds the following positions:

- *Managing Director of Alten Italia*
- *Chairman of ITA-STQB, the Italian Board of ISTQB®*
- *of ISTQB® Marketing Working Group*
- *Chairman of STF – Software Testing Forum, the most important event on testing held annually in Italy*

Andreas Ebbert-Karroum

leads the Agile Software Factory (ASF) at codecentric. For more than four years, he is a certified Scrum Master. Since then he could contribute his competencies in small and large (> 300 persons), internal and external, or local and global projects as developer, scrum master or product owner. Meanwhile, he's also a Professional Developer Scrum Trainer, conducting a hands-on training for Scrum team members, which codecentric has developed together with scrum.org and in which he shares with joy the knowledge gained in the ASF. More than 15 years ago, his interest in JavaSE/EE awakened, and it never vanished since then. His focus at codecentric is the continuous improvement of the development process in the Agile Software Factory, where technical, organizational and social possibilities are the challenging, external determining factors.

Paul Gerrard

is a consultant, teacher, author, webmaster, programmer, tester, conference speaker, rowing coach and most recently a publisher. He has conducted consulting assignments in all aspects of software testing and quality assurance, specialising in test assurance. He has presented keynote talks and tutorials at testing conferences across Europe, the USA, Australia, South Africa and occasionally won awards for them.

Educated at the universities of Oxford and Imperial College London, he is a Principal of Gerrard Consulting Limited and is the host of the UK Test Management Forum.

Mieke Gevers

In the IT industry for more than twenty years, Mieke Gevers has developed a special interest in the techniques and processes relating to performance management and automated testing. Mieke is a regular speaker at conferences and organizations worldwide. She served on the Program Committee for the EuroSTAR conference in 2007 and 2009 and was Program Chair of the Belgium Testing Days 2011 and 2012. A co-founder of the Belgian Testers Organization, Mieke has been a board member of KVIV and the ISTQB-affiliate Belgian Software Testing Qualifications Board.

Hans Schaefer

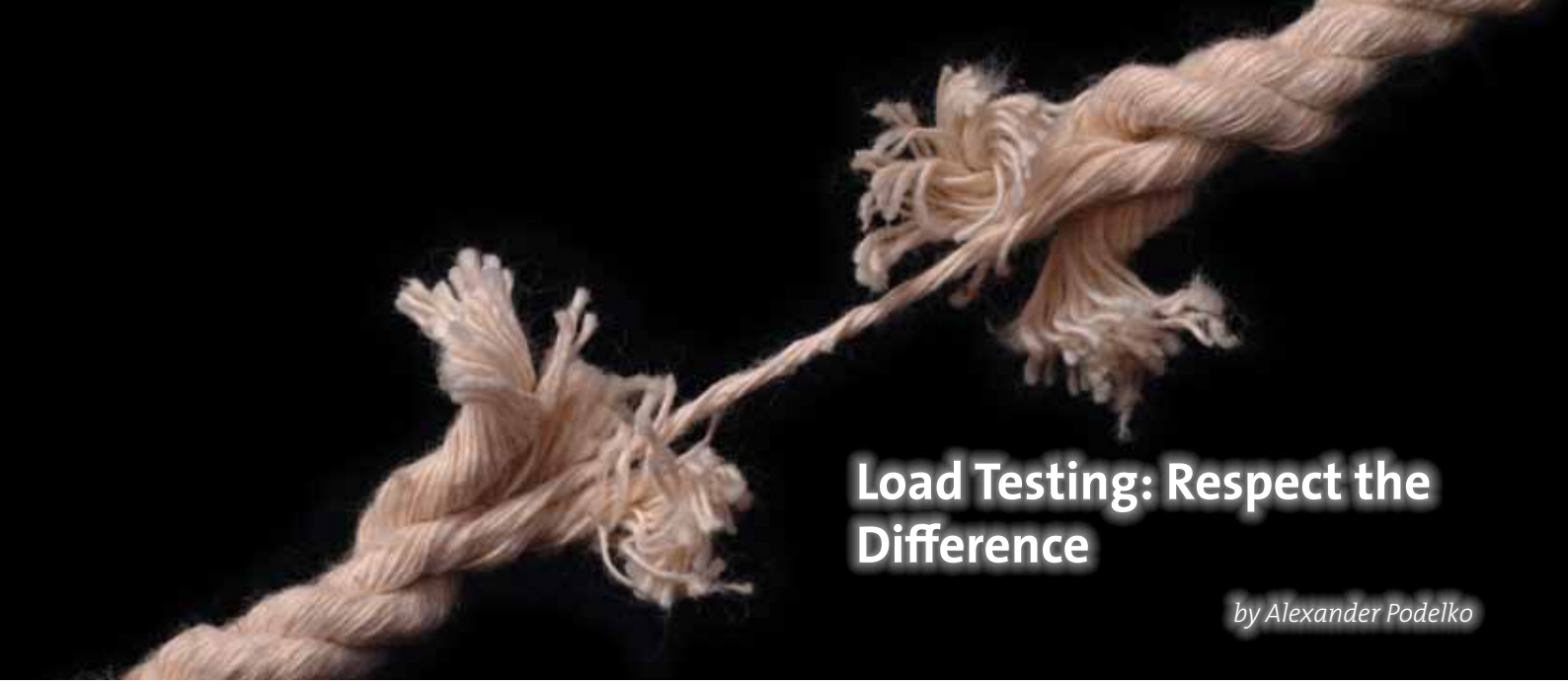
- *57 years old. Specialist in software testing*
- *Independent consultant in software testing and testing improvement matters. Guest lectures at several universities in Norway about Quality Assurance and Software Testing. Public and inhouse seminars in Software Review and Testing in Scandinavian and European countries. Regularly speaking at conferences. Several best paper awards.*

  (Best presentation award at CONQUEST 2003, best paper in 2004).
- *Test coordinator, ISEB Certified Tester (Foundation Level), ISTQB Certified Tester (Foundation and Advanced Levels) Chairman of Norwegian Testing Board.*
- *Active participant in running museum trains in Norway, certified for safety critical services on steam locomotives.*
- *M. Eng. from Technical University of Braunschweig, Germany.*
- *Computer science, railway signaling.*
- *Development of real time process control software at the Fraunhofer-Institute in Karlsruhe, Germany.*
- *Work with Center for Industrial Research in Oslo, Norway.*

- *Development of parts of an IDE-tool (Systemator). Later developing test tools and leader of quality improvement program. Consulting and lecturing about software quality.*

Yaron Tsubery
has been working in software since 1990, and has more than 20 years in testing field as a Test Engineer, Testing TL, and Testing Manager. His original profession was Systems Analyst.
Yaron Tsubery is the current CEO and founder of Smartest Technologies Ltd and has been a Director of QA & Testing Manager at Comverse since 2000 until 2010. Yaron was in charge of a large and distributed group that includes Testing Team Leaders and Test Engineers. The group deals with Functional, Non-functional, Load & Performance tests, some of which are done off-shore. The group has much experience in managing and controlling acceptance tests. Yaron is also known as a manager who has built testing groups in his past and also acted as a consultant to testing managers while they were building their testing groups.
Yaron worked in Product Management, Project Management and Development for at least 3 years before becoming a Director of QA & Testing.
Yaron has wide experience in banking business processes. For the last 10 years he has implemented best practices in a field where he is in charge of producing complex systems for telecommunication companies such as Vodafone, T-Mobile, NTT DoCoMo, Verizon Wireless etc'; Main systems in Mobile technology: i-mode, Mobile GateWay and Billing Proxy; in System Integration projects: Content Data Management solutions including OSS & BSS sub-systems, Video Solutions, IPTV and IMS.
Yaron is the current President of ISTQB® (International Software Testing Qualifications Board – www.istqb.org) and is also the President and founder of the ITCB (Israeli Testing Certification Board – www.itcb.org.il). He is a member of IQAMF (Israeli QA Managers Forum) and in SIGiST Israel.
Yaron has been invited as a speaker to some important international conferences to lecture on subjects related to testing, as well as he wrote articles that were published in professional magazines.

Erik van Veenendaal
(www.erikvanveenendaal.nl) is a leading international consultant and trainer, and a widely recognized expert in the area of software testing and quality management with over 20 years of practical testing experiences. He is the founder of Improve Quality Services BV (www.improveqs.nl) and one of the core developers of the TMap testing methodology and the TMMi test improvement model. Erik is a regular speaker both at national and international testing conferences and holds the EuroSTAR record winning the best tutorial award three times! In 2007 he received the European Testing Excellence Award for his outstanding contribution to the testing profession over the years. He has been working as a test manager and consultant in various domains, including finance, government and embedded software. He has written numerous papers and a number of books, including "Risk-based Testing: The PRISMA Approach", "ISTQB Foundations of Software Testing" and "The Little TMMi". Erik is also a former part-time senior lecturer at the Eindhoven University of Technology, vice-president of the International Software Testing Qualifications Board (2005–2009) and currently vice chair of the TMMi Foundation.



Load Testing: Respect the Difference

by Alexander Podelko

Talking about Test Centers of Excellence, it is important to remember that the skills and processes needed for load testing are quite different from functional testing. Often that results in creating a separate Performance Center of Excellence, but regardless of specific organizational structure, it is important to understand and respect the difference.

This article tries to contrast load testing with functional testing and highlights points that are often missed by people moving into load testing from functional testing or development. Applying the best practices and metrics of functional testing to load testing quite often results in disappointments, unrealistic expectations, sub-optimal test planning and design, and misleading results. While people who were involved in load testing or performance analysis may find many statements below to be trivial, it still may be beneficial to highlight the differences when we discuss processes and skills needed for Test Centers of Excellence.

Testing multi-user applications under realistic as well as stress loads remains the main way to ensure appropriate performance and reliability in production. There are many terms defining such kind of testing: load, performance, stress, scalability, concurrency, reliability, and many others. There are different (and sometimes conflicting) definitions of these terms, and these terms describe testing from somewhat different points of view, so they are not mutually exclusive.

While each kind of performance testing may be somewhat different, in most cases they use the same approach: applying multi-user synthetic workload to the system. We mostly use the term "load testing" further in that article because we try to contrast multi-user load testing with single-user functional testing. Everything mentioned here applies to performance, stress, scalability, reliability and other kinds of testing as far as these features are tested by applying load.

Load Testing Process Overview

Load testing is emerging as an engineering discipline of its own, based on "classic" testing from one side, and system performance analysis from another side. A typical load testing process is shown in figure 1.

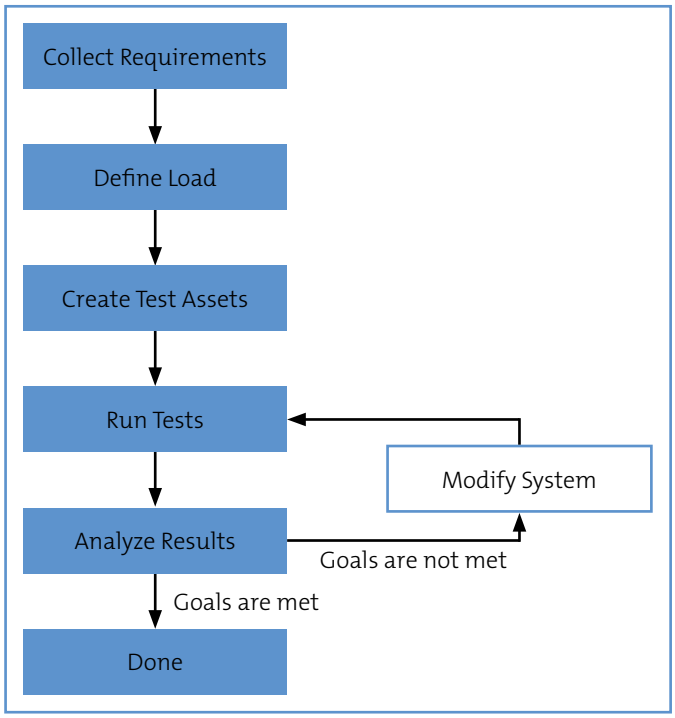
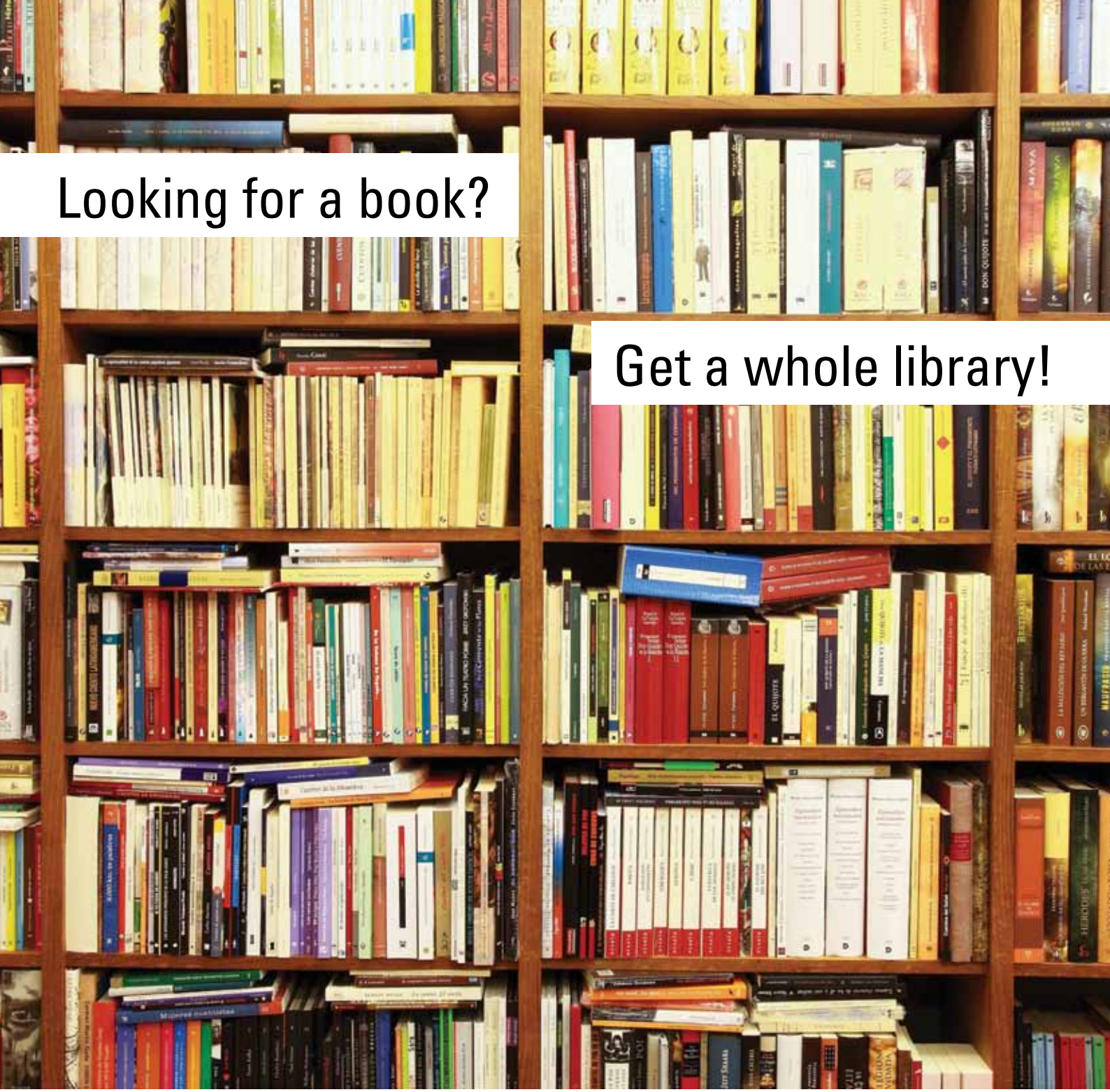


Fig.1 Load testing process

We explicitly define two different steps here: "define load" and "create test assets". The "define load" step (sometimes referred to as workload characterization or workload modeling) is a logical description of the load we want to apply (like "that group of users login, navigate to a random item in the catalog, add it to the shopping cart, pay, and logout with average 10 seconds think time between actions"). The "create test assets" step is the implementation of this workload, and conversion of the logical description into something that will physically create that load during the "run tests" step. While for manual testing that can be just the description given to each tester, usually it is something else in load testing – a program or a script.

Quite often load testing goes hand-in-hand with tuning, diagnostics, and capacity planning. It is actually represented by the back loop on the fig.1: if we don't meet our goal, we need optimize the system to improve performance. Usually the load testing process implies tuning and modification of the system to achieve the goals.



iSQI® is your one-stop-agency for learning resources, professional trainings for the most important international standards and the most convenient certification services in the world.

Every exam scheduled between July 1st and September 30th 2012 not only allows you to enjoy our renowned certification examination services. In addition, you will get free access to the Learntesting e-book Library, the best place for professional literature on testing, QA and related subjects on the net!

Register your certification today at https://www.isqi.org/en/registration.html or contact us at info@isqi.org

learntesting

www.isqi.org
CERTiFYING PEOPLE

iSQI®
International Software Quality Institute

Load testing is not a one-time procedure. It spans through the whole system development life cycle. It may start from technology or prototype scalability evaluation, continue through component / unit performance testing into system performance testing before deployment and follow up in production (to troubleshooting performance issues and test upgrades / load increases).

What to Test

Even in functional testing we have a potentially unlimited number of test cases and the art of testing is to choose a limited set of test cases that should check the product functionality in the best way with given resource limitations. It is much worse with load testing. Each user can follow a different scenario (a sequence of functional steps), and even the sequence of steps of one user versus the steps of another user could affect the results significantly.

Load testing can't be comprehensive. Several scenarios (use cases, test cases) should be chosen. Usually they are the most typical scenarios, the ones that most users are likely to follow. It is a good idea to identify several classes of users – for example, administrators, operators, users, or analysts. It is simpler to identify typical scenarios for a particular class of users. With that approach, rare use cases are ignored. For example, many administrator-type activities can be omitted as far as there are few of them compared with other activities.

Another important criterion is risk. If a "rare" activity presents a major inherent risk, it can be a good idea to add it to the scenarios to test. For example, if database backups can significantly affect performance and should be done in parallel with regular work, it makes sense to include a backup scenario in performance testing.

Code coverage usually doesn't make much sense in load testing. It is important to know what parts of code are being processed in parallel by different users (that is almost impossible to track), not that particular code path was executed. Perhaps it is possible to speak about "component coverage", making sure that all important components of the system are involved in performance testing. For example, if different components are responsible for printing HTML and PDF reports, it is a good idea to add both kinds of printing to testing scenarios.

Requirements

In addition to functional requirements (which are still valid for performance testing – the system still should do everything it is designed to do under load), there are other classes of requirements:

- Response times – how fast the system handles individual requests or what a real user would experience
- Throughput – how many requests the system can handle
- Concurrency – how many users or threads can work simultaneously

All of them are important. Good throughput with long response times often is as unacceptable as good response times, but just for a few users.

Acceptable response times should be defined in each particular case. A response time of 30 minutes may be excellent for a big batch job, but it is absolutely unacceptable for accessing a Web page for an online store. Although it is often difficult to draw the line here, it is rather a usability or common sense decision. Keep in mind that for multi-user testing we get multiple response times for each transaction, so we need to use some aggregate values like averages or percentiles (for example, 90% of response times are less than this value).

Throughput defines the load on the system. Unfortunately, quite often the number of users (concurrency) is used to define the load for interactive systems instead of throughput. Partially because that number is often easier to find, partially because it is the way how load testing tools define load. Without defining what each user is doing and how intensely (i.e. throughput for one user), the number of users is not a good measure of load. For example, if there are 500 users running short queries each minute, we have throughput of 30,000 queries per hour. If the same 500 users are running the same queries, but one per hour, the throughput is 500 queries per hour. So with the same 500 users we have a 60-fold difference between loads and, probably, at least 60-fold difference in hardware needed.

The intensity of load can be controlled by adding delays (often referred as "think time") between actions in scripts or harness code. So one approach is to start with the total throughput the system should handle, then find the number of concurrent users, get the number of transactions per user for the test, and then try to set think times to ensure the proper number of transactions per user.

Finding the number of concurrent users for a new system can be tricky too. Usually information about real usage of similar systems can help to make an initial estimate. Another approach may be to assume what share of named (registered in the system) users are active (logged on). So if that share is 10%, 1,000 named users results in 100 active users. These numbers, of course, depend greatly on the nature of the system.

Workload Implementation

If we work with a new system and have never run a load test against it before, the first question is how to create load. Are we going to generate it manually, use a load testing tool, or create a test harness?

Manual testing could sometimes work if we want to simulate a small number of users. However, even if it is well organized, manual testing will introduce some variation in each test, making the test difficult to reproduce. Workload implementation using a tool (software or hardware) is quite straightforward when the system has a pure HTML interface, but even if there is an applet on the client side, it may become a serious research task, not to mention dealing with proprietary protocols. Creating a test harness requires more knowledge about the system (for example, an API) and some programming skills. Each choice requires different skills, resources, and investments. Therefore, when starting a new load-testing project, the first thing to do is to decide how the workload will be implemented and to check that this way will really work. After we decide how to create the workload, we need to find a way to verify that the workload is really being applied.

Workload Verification

Unfortunately, an absence of error messages during a load test does not mean that the system works correctly. An important part of load testing is workload verification. We should be sure that the applied workload is doing what it is supposed to do and that all errors are caught and logged. The problem is that in load

Agile Dev Practices

March 4–6, 2013 in Düsseldorf, Germany
The Conference for Agile Developers

Call for Papers

The call for papers ends by August 19, 2012.
The theme of this year's conference is
"Retrospectives on Agile – Forward Reasoning".

Go to **www.agiledevpractices.com** and submit your paper!

Topics of interest:

The values and principles behind the agile practices · Clean code · Refactoring · Dealing with legacy code · Version Control Systems · Users stories and acceptance criteria · BDD/ATDD/TDD · Meetings · Retrospectives · Product backlog management and estimation · Practices for adoption · Agile adoption stories · What worked and what didn't · DevOps · Distributed teams management · Roles in the team

www.agiledevpractices.com

testing we work on the protocol or API level and often don't have any visual clues that something doesn't work properly. Workload can be verified directly (by analyzing server responses) or, in cases where this is impossible, indirectly (for example, by analyzing the application log or database for the existence of particular entries).

Many tools provide some ways to verify workload and check errors, but you need understanding what exactly is happening. Many tools report only HTTP errors for Web scripts by default (such as 500 "Internal Server Error"). If we rely on the default diagnostics, we could still believe that everything is going well when we are actually getting "out of memory" errors instead of the requested reports. To catch such errors, we should add special commands to our script to check the content of HTML pages returned by the server.

The Effect of Data

The size and structure of data could affect load test results drastically. Using a small sample set of data for performance tests is an easy way to get misleading results. It is very difficult to predict how much the data size affects performance before real testing. The closer the test data is to production data, the more reliable are test results.

Running multiple users hitting the same set of data (for example, playback of an automatically created script without proper modifications) is an easy way to get misleading results. This data could be completely cached, and we will get much better results than in production. Or it could cause concurrency issues, and we will get much worse results than in production. So scripts and test harnesses usually should be parameterized (fixed or recorded data should be replaced with values from a list of possible choices) so that each user uses a proper set of data. The term "proper" here means different enough to avoid problems with caching and concurrency, which is specific for the system, data, and test requirements.

Another easy trap with data is adding new data during the tests without sufficient considerations. Each new test will create additional data, so each test would be done with a different amount of data. One way of running such tests is to restore the system to the original state after each test or group of tests. Or additional tests can be performed to prove that a change of data volume inside a specific range does not change the outcome of that particular test.

Exploring the System

At the beginning of a new project, it is a good practice to run some tests to figure out how the system behaves before creating formal plans. If no performance tests have been run, there is no way to predict how many users the system can support and how each scenario will affect overall performance. Modeling can help here to find the potential limits, but a bug in the code or an environmental issue can dwarf scalability.

It is good to check that we do not have any functional problems. Is it possible to run all requested scenarios manually? Are there any

performance issues with just one or with several users? Are there enough computer resources to support the requested scenarios? If we have a functional or performance problem with one user, usually it should be fixed before starting performance testing with that scenario.

Even if there are extensive plans for performance testing, an iterative approach will fit better here. As soon as a new script is ready, run it. This will help to understand how well the system can handle a specific load. The results we get can help to improve plans and discover many issues early. By running tests we are learning the system and may find out that the original ideas about the system were not completely correct. A "waterfall" approach, when all scripts are created before running any multi-user test, is dangerous here: issues would be discovered later and a lot of work may need to be redone.

Assumed Activities

Usually when people talk about performance testing, they do not separate it from tuning, diagnostics, or capacity planning. "Pure" performance testing is possible only in rare cases when the system and all optimal settings are well known. Usually some tuning activities are necessary at the beginning of testing to be sure that the system is properly tuned and the results will be meaningful. In most cases, if a performance or reliability problem is found, it should be diagnosed further until it becomes clear how to handle it. Generally speaking, performance testing, tuning, diagnostics, and capacity planning are quite different processes, and excluding any of them from the test plan if they are assumed will make the plan unrealistic from the beginning.

Test Environment

Conducting functional testing in virtualized and cloud environments is quite typical and has many advantages. While many companies promote load testing in the cloud (or from the cloud), it makes sense only for certain types of load testing. For example, it should work fine if we want to test how many users the system supports, would it crash under load of X users, how many servers we need to support Y users, etc., but are not too concerned with exact numbers or variability of results (or even want to see some real-life variability).

However, it doesn't quite work for performance optimization, when we make a change in the system and want to see how it impacts performance. Testing in a virtualized or cloud environment with other tenants intrinsically has some results variability as far as we don't control other activities and, in the cloud, usually don't even know the exact hardware configuration.

So when we talk about performance optimization, we still need an isolated lab in most cases. And, if the target environment for the system is a cloud, it probably should be an isolated private cloud with hardware and software infrastructure similar to the target cloud. And we need monitoring access to underlying hardware to see how the system maps to the hardware resources and if it works as expected (for example, testing scaling out or evaluating impacts to/from other tenants – which probably should be one more kind of performance testing to do).

Time Considerations

Performance tests usually take more time than functional tests. Usually we are interested in the steady mode during load testing. It means that all users need to log in and work for some time to be sure that we see a stable pattern of performance and resource utilization. Measuring performance during transition periods can be misleading. The more users we simulate, the more time we will usually need to get into the steady mode. Moreover, some kinds of testing (reliability, for example) can require a significant amount of time – from several hours to several days or even weeks. Therefore, the number of tests that can be run per day is limited. It is especially important to consider this during tuning or diagnostics, when the number of iterations is unknown and can be large.

Simulating real users requires time, especially if it isn't just repeating actions like entering orders, but a process when some actions follow others. We can't just squeeze several days of regular work into fifteen minutes for each user. This will not be an accurate representation of the real load. It should be a slice of work, not a squeeze.

In some cases we can make the load from each user more intensive and respectively decrease the number of users to keep the total volume of work (the throughput) the same. For example, we can simulate 100 users running a small report every five minutes instead of 300 users running that report every fifteen minutes. In this case, we can speak about the ratio of simulated users to real users (1:3 for that example). This is especially useful when we need to perform a lot of tests during the tuning of the system or trying to diagnose a problem to see the results of changes quickly. Quite often that approach is used when there are license limitations.

Still "squeezing" should only be used in addition to full-scale simulation, not instead of it. Each user consumes additional resources for connections, threads, caches, etc. The exact impact depends on the system implementation, so simulation of 100 users running a small report every ten minutes doesn't guarantee that the system will support 600 users running that report every hour. Moreover, tuning for 600 users may differ significantly from tuning for 100 users. The higher the ratio between simulated and real users, the more is the need to run a test with all users to be sure that the system supports that number of users and that the system is properly tuned.

Testing Process

Three specific characteristics of load testing affect the testing process and often require closer work with development to fix problems than functional testing does. First, a reliability or performance problem often blocks further performance testing until the problem is fixed or a workaround is found. Second, usually the full setup, which often is very sophisticated, should be used to reproduce the problem. However, keeping the full setup for a long time can be expensive or even impossible. Third, debugging performance problems is a sophisticated diagnostic process that usually requires close collaboration between a performance engineer running tests and analyzing the results and a developer profiling and altering code. Special tools may be needed: many tools, such as regular debuggers and profilers, work fine in a single-user environment, but do not work in the multi-user environment due to huge performance overheads.

These three characteristics make it difficult to use an asynchronous process in load testing (which is often used in functional testing: testers look for bugs and log them into a defect tracking system, and then the defects are prioritized and independently fixed by development). What is often required is the synchronized work of performance engineering and development to fix the problems and complete performance testing.

A Systematic Approach to Changes

The tuning and diagnostic processes consist of making changes in the system and evaluating their impact on performance (or problems). It is very important to take a systematic approach to these changes. This could be, for example, the traditional approach of "one change at a time" (sometimes referred as "one factor at a time", or OFAT) or using the design of experiments (DOE) theory. "One change at a time" here does not imply changing only one variable; it can mean changing several related variables to check a particular hypothesis.

The relationship between changes in the system parameters and changes in the product behavior is usually quite complex. Any assumption based on common sense could be wrong. A system's reaction under heavy load could differ drastically of what was expected. So changing several things at once without a systematic approach will not give the understanding of how each change affects results. This could mess up the testing process and lead to incorrect conclusions. All changes and their impacts should be logged to allow rollback and further analysis.

Result Analysis

Load testing results bring much more information than just passed/failed. Even if we do not need to tune the system or diagnose a problem, we usually should consider not only transaction response times for all different transactions (usually using aggregating metrics such as average response times or percentiles), but also other metrics such as resource utilization. The systems used to log functional testing results usually don't have much to log all this information related to load testing results.

Result analysis of load testing for enterprise-level systems can be quite difficult and should be based on a good knowledge of the system and its performance requirements, and it should involve all possible sources of information: measured metrics, results of monitoring during the test, all available logs and profiling results (if available). We need information not only for all components of the system under test, but also for the load generation environment. For example, a heavy load on load generator machines can completely skew results, and the only way to know that is to monitor those machines.

There is always a variation in results of multi-user tests due to minor differences in the test environment. If the difference is large, it makes sense to analyze why and adjust tests accordingly. For example, restart the program, or even reboot the system, before each test to eliminate caching effects.

Summary

To wrap up, there are serious differences in processes and required skills between load and functional testing. Some of them were discussed in this article, but these are rather examples than a comprehensive list. It is important to understand these differences when talking about Test Centers of Excellence, regardless of specific organizational structures: while load and functional testing both are testing, trying to fit them into the same organizational structure without consideration of their specifics may be problematic.

> biography

Alexander Podelko

For the last fifteen years Alex Podelko has worked as a performance engineer and architect for several companies. Currently he is Consulting Member of Technical Staff at Oracle, responsible for performance testing and optimization of Hyperion products. Alex serves as a director for the Computer Measurement Group (CMG) http://cmg.org, a volunteer organization of performance and capacity planning professionals. He blogs at http://alexanderpodelko.com/blog and can be found on Twitter as @apodelko.

Enjoy **Test Case Design** in the Cloud

CaseMaker SaaS supports systematically the test cases design by covering the techniques taught in the "ISTQB® Certified Tester program" and standardized within the British Standard BS 7925. The implemented techniques are: Equivalence Partitioning, Boundary Check, Error Guessing, Decision Tables and Pairwise Testing. Furthermore Risk Based Testing is supported.

CaseMaker SaaS fits between Requirement Management and Test Management/Test Automation.

Subscribe and try for free! Decide afterwards.

One license starts at

75€/month (+ VAT)

http://saas.casemaker.eu

PayPal

CaseMaker® SAAS

Managing Technical Test Debt

A Critical Success Factor in Agile Delivery

by Bob Galen

I've been programming and testing for over a quarter of a century. One of the things I like about the agile methods is that they gave a name to something that developers always had to deal with – Technical Debt. Everyone knew about it, but not having a representative term sort of lessened its impact or import. It was just so easy to ignore if we didn't know what to call it.

You know what I mean...

Technical debt is that sinking feeling that developers have when trying to change old, brittle, poorly maintained code. When they draw straws and the short straw gently prays that they can get over just one more duct-taped repair to the codebase without it falling apart. Where team members mumble softly to themselves during a build – please, please don't fail.

Fast forward to the agile methodologies and we've quantified it. We try to attack it with other techniques like Test Driven Development (TDD) and refactoring and speak about incremental development, working in end-to-end slices, where we can mitigate technical debt and properly clean up after ourselves.

In this article I want to shift the perspective from development or code-centric technical debt and draw a correlation between it and testing. You see, I think testers and software testing suffer from the same sort of problems, but nobody is really talking about it in the same way. I want to introduce the term Technical Test Debt (TTD) as the testing equivalent to technical debt – so let's explore that notion a bit further.

Technical Test Debt

Test debt is more broadly nuanced than its development counterpart. The most direct comparison is to test cases that fall out of repair. I've always felt that test cases are the testing equivalent to lines of code from a deliverable perspective. Additionally, they both have similar design and creation characteristics.

So, technical test debt can be created where testers have failed to keep their test case documentation (assuming manual in this case) up-to-date with the evolution of the product or project. But that's not the only contributor to TTD. Here's a list that defines some of the testing activities and/or artifacts that, if done poorly or deferred, inevitably begin to erode the testing integrity of a software project –

TTD in Artifacts

- Undefined or unkempt test cases – dated, not matching current software functionality
- Badly executed exploratory testing that kept poor documentation surrounding session results
- Lack of collaborative test design – strategies for testing challenging project details are lost or non-existent
- Lack of collaborative user story writing w/o acceptance tests
- Lack of acceptance tests or user story maintenance and archival

TTD in Automation

- Badly designed test automation – prone to break, not tolerant of application changes w/o large-scale maintenance; hard to extend for new functionality
- Not keeping up with test automation maintenance – either at a framework or test case perspective as the AUT evolves
- Automated application coverage actually declining over time

TTD in Collaboration and Support

- Developers not doing their part in keeping up with quality work (unit testing, design focus, inspection focus, and an overall quality focus) within the team
- Ratio issues (and the team not doing their part) – potentially agile teams without or with insufficient numbers of testers – and the team not able or willing to pick up the slack

TTD Commitment to Quality

- Tests that aren't run during each sprint or iteration due to a lack of time or because of planned risk-based testing
- Deferred major test intervals; more focused towards non-functional tests
- Testers aren't even assigned to agile teams
- Not taking a "whole team" view towards quality and investing in code reviews and inspections, root cause analysis, and continuous improvement

Agile TESTING DAYS

November 19–22, 2012 in Potsdam, Germany

Dorint Hotel Sanssouci Berlin/Potsdam

www.agiletestingdays.com

Become Agile Testing Days Sponsor 2012

The Agile Testing Days 2012, from November 19–22 is an annual European conference for and by international professionals involved in the agile world. Since 2009 we got a lot of addition to our agile family and we are happy to present the 4th edition of the Europeans greatest agile event of the year. What you will expect:

- 4 inspiring conference days
- 9 stunning keynotes
- 10 instructive tutorials
- 12 informative sponsor sessions
- 40 amazing talks
- More than 70 speakers
- Over 400 testers from all over the world
- And an exhibition area of 500sqm

Tutorials

Time	Tutorial
08:00–09:00	Registration
09:00–17:30	**"Management 3.0 Workshop"** Jurgen Appelo
09:00–17:30	**"Making Test Automation Work in Agile Projects"** Lisa Crispin
09:00–17:30	**"Transitioning to Agile Testing"** Janet Gregory:
09:00–17:30	**"Introduction to Disciplined Agile Delivery"** Scott W. Ambler
09:00–17:30	**"Beheading the legacy beast"** Ola Ellnestam
09:00–17:30	**"Fully integrating performance testing into agile development"** Scott Barber
09:00–17:30	**"Mindful Team Member: Working Like You Knew You Should"** Lasse Koskela
09:00–17:30	**"Mind Maps: an agile way of working"** Huib Schoots & Jean-Paul Varwijk
09:00–17:30	**"Winning big with Specification by Example: Lessons learned from 50 successful projects"** Gojko Adzic
09:00–17:30	**"Software Testing Reloaded – So you wanna actually DO something? We've got just the workshop for you. Now with even less powerpoint!"** Matt Heusser & Pete Walen

All tutorials include two coffee breaks (11:00 and 15:30) and lunch (13:00–14:00).

November 19–22, 2012 in Potsdam, Germany
Dorint Hotel Sanssouci Berlin/Potsdam
www.agiletestingdays.com

Agile TESTING DAYS

Conference Day 1

Time	Track 1	Track 2	Track 3	Track 4	Vendor Track
08:00–09:20	Registration				
09:20–09:25	Opening				
09:25–10:25	**Keynote: "Disciplined Agile Delivery: The Foundation for Scaling Agile"** – Scott W. Ambler				
10:25–10:30	Break				
10:30–11:15	**"5A – assess and adapt agile activities"** Werner Lieblang & Arjan Brands	**"Moneyball and the Science of Building Great Agile Team"** Peter Varhol	**"Get them in(volved)"** Arie van Bennekum	**"Testing distributed projects"** Hartwig Schwier	
11:15–11:40	Break				
11:40–12:25	**"The Agile Manifesto Dungeons: Let's go really deep this time!"** Cecile Davis	**"Balancing and growing agile testing with high productive distributed teams"** Mads Troels Hansen & Oleksiy Shepetko	**"We Line Managers Are Crappy Testers – Can We Do Something About It"** Ilari Henrik Aegerter	**"The many flavors and toppings of exploratory testing"** Gitte Ottosen	
12:25–13:45	Lunch				
13:45–14:45	**Keynote: "Myths About Agile Testing, De-Bunked"** – Janet Gregory & Lisa Crispin				

Time	Track 1	Track 2	TBD	TBD	TBD	Vendor Track
14:45–16:15	**Consensus Talks** 10 min. each	**Open Space** Cirilo Wortel	**TestLab** Bart Knaack & James Lyndsay	**Testing Dojos**	**Coding Dojos** Meike Mertsch & Michael Minigshofer	**Product Demo**

Time	Track 1	Track 2	Track 3	Track 4	Vendor Track
16:15–16:40	Break				
16:40–17:25	**"The Beating Heart of Agile"** Andrea Provaglio	**"Why World of Warcraft is like being on an agile team, when it isn't and what we can learn from online role playing games"** Alexandra Schladebeck	**"Agile communication: Back and forth between managers and teams"** Zuzana Sochova & Eduard Kunce	**"Developers Exploratory Testing – Raising the bar"** Sigge Birgisson	
17:25–17:30	Break				
17:30–18:30	**Keynote: "Self Coaching"** – Lasse Koskela				
19:00	Social Event				

Conference Day 2

Time	Track 1	Track 2	Track 3	Track 4	Vendor Track
07:30–09:20	Registration				
08:10–08:55	**Early Keynote: TBD**				
09:20–09:25	Opening				
09:25–10:25	**Keynote: "How to change the world"** – Jurgen Appelo				
10:25–10:30	Break				
10:30–11:15	**"Continuous Delivery of Long-Term Requirements"** Paul Gerrard	*TBD*	**"How releasing faster changes testing"** Alexander Schwartz	**"Testers are bearers of good news"** Niels Malotaux	
11:15–11:40	Break				
11:40–12:25	**"Experiences with introducing a Continuous Integration Strategy in a Large Scale Development Organization"** Simon Morley	**"Skills & techniques in the modern testing age"** Rik Teuben	**"Continuous Delivery: from dream to reality"** Clement Escoffier	**"Ten qualities of an agile test-oriented developer"** Alexander Tarnowski	
12:25–13:45	Lunch				
13:45–14:45	**Keynote: "Adaptation and Improvisation – but your weakness is not your technique"** – Markus Gärtner				

Time	Track 1	Track 2	TBD	TBD	TBD	Vendor Track
14:45–16:15	**Consensus Talks** 10 min. each	**Open Space** Cirilo Wortel	**TestLab** Bart Knaack & James Lyndsay	**Testing Dojos**	**Coding Dojos** Meike Mertsch & Michael Minigshofer	**Product Demo**

Time	Track 1	Track 2	Track 3	Track 4	Vendor Track
16:15–16:40	Break				
16:40–17:25	**"From CI 2.0+ to Agile ALM"** Michael Hüttermann	**"Testers Agile Pocketbook"** Stevan Zivanovic	**"Extending Continuous Integration and TDD with Continuous Testing"** Jason Ayers	**"Excelling as an Agile Tester"** Henrik Andersson	
17:25–17:30	Break				
17:30–18:30	**Keynote: "Reinventing software quality"** – Gojko Adzic				

Agile TESTING DAYS

Conference Day 3

Time	Track 1	Track 2	Track 3	Track 4	Vendor Track
07:30–09:10	Registration				
08:10–08:55	**Early Keynote: TBD**				
09:10–09:15	Opening				
09:15–10:15	**Keynote: "Fast Feedback Teams"** – Ola Ellnestam				
10:15–10:20	Break				
10:20–11:05	**"Exceptions, Assumptions and Ambiguity: Finding the truth behind the Story"** David Evans	**"BDD with Javascript for Rich Internet Applications"** Carlos Blé & Ivan Stepániuk	**"Automation of Test Oracle – unachievable dream or tomorrow's reality"** Dani Almog	**"You Can't Sprint All the time – the importance of slack"** Lloyd Roden	
11:05–11:30	Break				
11:30–12:15	**"Combining requirements engineering and testing in agile"** Jan Jaap Cannegieter	**"TDD-ing Javascript Front Ends"** Patrick Kua	**"Archetypes and Templates: Building a lean, mean BDD automation machine for multiple investment platforms"** Mike Scott & Tom Roden	**"Taking over a bank with open source test tooling"** Cirilo Wortel	
12:15–13:00	**"Agile Solutions – Leading with Test Data Management"** Ray Scott	**"Changing Change"** Tony Bruce	**"Technical Debt"** Thomas Sundberg	**"Changing the context: How a bank changes their software development methodology"** Huib Schoots	
13:00–14:10	Lunch				
14:10–15:10	**Keynote: "The ongoing evolution of testing in agile development"** – Scott Barber				
15:10–15:15	Break				
15:15–16:00	**"Thinking and Working Agile in an Unbending World"** Peter Walen	**"Sprint Backlog in ATDD"** Ralph Jocham	**"Mobile test automation at mobile scale"** Dominik Dary & Michael Palotas	**"Quality On Submit, Continuous Integration in Practice"** Asaf Saar	
16:00–16:05	Break				
16:05–17:05	**Keynote: "The Great Game of Testing"** – Matt Heusser				
17:05	Closing				

Please note that the program is subject to change.

Agile TESTING DAYS

November 19–22, 2012 in Potsdam, Germany
Dorint Hotel Sanssouci Berlin/Potsdam
www.agiletestingdays.com

Become Agile Testing Days Sponsor 2012

For this phenomenal event, we are looking for supporters.
Please have a look at our portfolio and create your own sponsorship package:

Exhibitor

Diamond Exhibitor:	16,200.00 €*
Platinum Exhibitor:	10,800.00 €*
Gold Exhibitor:	5,400.00 €*
Silver Exhibitor:	3,600.00 €*

Conference Sponsor

MIATPP Award Trophy Sponsor:	2,970.00 €*
Conference Bag Insert:	495.00 €*
Coffee Break Cup Sponsor:	1,530.00 €*
Social Event Sponsor:	8,990.00 €*

Media Sponsor

Program First Page Advert:	990.00 €*
Program Full Page Advert:	495.00 €*
Program Half page Advert:	249.00 €*

Session Sponsor

90 Minutes Product Demo:	2,250.00 €*
45 Minutes Early Keynote:	2,250.00 €*

* Early Bird price – valid until July 31, 2012.

Please find details on packages online at www.agiletestingdays.com

Exhibitors & Supporters 2011:

T-Systems · Seapine Software · NOKIA · XING · SEQIS The Austrian Software Test Experts! · telerik deliver more than expected
dpunkt.verlag · hp · iSQI · POTSDAM · JetBRAINS Develop with pleasure! · testing experience · Agile Record · Díaz Hilterscheid

A Díaz & Hilterscheid Conference

Díaz & Hilterscheid Unternehmensberatung GmbH
Kurfürstendamm 179
10707 Berlin
Germany

Phone: +49 (0)30 74 76 28-0
Fax: +49 (0)30 74 76 28-99

info@agiletestingdays.com
www.agiletestingdays.com
www.xing.com/net/agiletesting
www.twitter.com/AgileTD

I've categorized them into areas just to focus the discussion. There's nothing special about the categorization, nor do I consider the list to be exhaustive – meaning, I suspect there are "other" kinds of test debt out there to worry about.

So, technical test debt is often the case where we've become lazy in our tidying up of our testing artifacts and activities of all kinds. That laziness can literally be just that, team members who ignore this work. The debt can also be driven by other factors, which I'll explore next.

Root Causes

TTD primarily happens because of several factors; probably lack of time being the most prevalent. Usually there are far more developers on agile projects than testers and this can create a natural skew in the teams' attention away from quality factors and mitigating TTD.

Another factor is driven from the business side. Quite often Product Owners and other stakeholders struggle with the true cost of software development. They like to consider only initial coding costs including minimal or initial testing in their views. Beyond this, they often try to ignore debt, whether software or TTD, as being irrelevant or a nuisance to the business – being purely something for the team to resolve on their own. So business funding and prioritization can be the inhibitor in this case.

There is another, much more subtle factor that can't be blamed on "them" or on leadership. One part of it is certainly skill based. Team members may simply not be trained in solid testing practices and the "why" behind managing TTD. Another part, however, is centered on the teams' overall commitment to using professional testing practices in their delivery of solid code. This is the sort of laziness I referred to earlier and it segues quite nicely into a similar problem in the agile development community.

Testing Craftsmanship

Along with others, 'Uncle' Bob Martin has recently been championing the notions of software craftsmanship and professionalism with respect to software development. I believe the same notions apply to our craft of testing.

At some fundamental level we, the testers within agile teams, **allow** TTD to creep into our projects. We don't make it visible to our teams and our stakeholders and we allow it to increase. While it frustrates us and reduces our overall productivity, we would rather choose to live with it as a normal course of events.

I guess an equivalent in software is developers hacking their code. Sure, it works now, but it incrementally makes everyone's job harder over time. And inevitably the team becomes less productive and creative; and from a business perspective – less competitive.

So the prime directive in battling TTD surrounds your **resolve to do things right the first time**. To increase your professionalism to not create it or, at the very least, make the compromises visible to your team and your agile stakeholders so that they can make a decision on how much is "good enough".

Exploring TTD Specifics

Now I want to go through the four core areas of technical test debt. Exploring each more fully in turn and looking more towards the specific drivers in each case while making recommendations how mature agile teams should defend against or battle against these debts.

If you recall, the four core TTD areas are:

1. Artifacts
2. Automation
3. Collaboration and Support
4. Commitment to Quality

Let's explore each in turn –

Artifacts

This is one of the more prevalent cases of TTD in many agile teams – primarily because the testers on the teams are usually viewed as being independently responsible for testing work – that work being clearly outside the whole teams' purview. There's an interesting anti-pattern I've seen where the teams are very focused on design and working code – even the quintessential sprint review being primarily focused on working code. Rarely does the team make their testing efforts transparent in the review. I think this is a huge mistake.

One place to turnaround this TTD problem area starts in sprint planning – asking teams to better represent test work in the same way that they represent designs and code. I'd even like to see the teams' Done-Ness requirements expanded to include TTD implications – ensuring it gets the right level of transparency.

Beyond this, the whole teams need to understand the value of their testing artifacts – of writing test cases, creating scripts, logging session results from exploratory testing sessions, etc. It's not a question of wasting time or being "un-agile", but more so focusing on creating appropriately leveled and sufficient documentation so that further testing can be easily quantified, repeated, and automated. Don't view this as waste or gold–plating. Instead, these are simply solid, context-based, and professional test practices.

And finally, show these results in your sprint reviews as they are part of your "working code" deliverables, and they should be exposed, highlighted, and appreciated!

Automation

Very few "entry level" agile teams have high degrees of test automation in place when they initially adopt the methodologies. Usually, they're dealing with legacy code of varying degrees of quality, a somewhat under-maintained repository of manual test cases, and some bits of UI-based automation that needs a lot of care and feeding. This represents their testing coverage for their application set.

Quite often this creates a tenuous situation where these teams and their customers or Product Owners minimize the importance of test automation. Instead the testers are simply consumed with mostly manually testing each sprint's results – so there is never the time for creating or maintaining automation.

A frequent mistake in these contexts is assuming that **only the testers** can write test automation. Nothing could be further from the truth! Establishing a team-based culture where literally anyone can write automation is a start to handling this debt.

Addressing automation debt is a simple exercise. It begins with the product backlog and the Product Owner's commitment to beating down their TTD. In this case, automating all new features is a great way to start – so that you don't continue to dig yourself deeper in debt.

Beyond new feature stabilization, you'll want to continue to increase your automation infrastructural investment and automation coverage. The truth is that it's your safety net for catching issues as you move more iteratively and quickly. So the team should look for high priority automation targets in their legacy codebase and place stories in the backlog for automating them.

Often organizations will reserve a percentage of their backlogs for this sort of activity – say 10-20% depending upon how much debt they're trying to clean up. The key point is to stay committed to automation until you hit your debt reduction targets.

Collaboration and Support

As I said above in the automation section, the business clearly needs to take a stand fighting TTD. It's as simple as that. If the business won't support the team's efforts controlling TTD, then it will continue to grow and incrementally undermine the effectiveness of the team. However, it goes beyond simple business support. The teams themselves need to play a strong role in helping stakeholders understand, quantify, estimate level of effort, and then effectively execute TTD reduction efforts.

How?

It begins with planning, both at the release and sprint levels, within the product backlog. Teams need to define backlog items or user stories that clearly target the debt. And you can't simply come to the business and ask to fix five years' worth of TTD all at once.

You need to collaborate with them; identifying critical areas and partnering with the business in prioritization. You also need to explain the impacts in terms they can understand – defect exposure, maintainability limitations, new feature development inhibition, and performance reductions are just a few relevant focus areas.

In addition, you need to articulate a clear strategy regarding TTD. It can't simply be statements like – "we're totally broken... and we need to fix everything now...". Instead you need to define a thoughtful strategy and make recommendations that connect to ongoing business strategies. For example, your strategy might align in this way –

- Initially stopping incremental TTD increases (stopping the insanity)
- Establishing baselines for improvement, quarterly targets that align with the existing product roadmap
- Incrementally, sprint over sprint and release over release, quantify improvement focus points via user stories with clear scope and clear acceptance tests
- Demonstrating the team's internal commitment to improvement at Sprint Reviews – showing results and improvement data trending wherever possible

This sort of holistic game planning, high degrees of transparency, and alignment to your specific challenges and business context goes a long way in gaining business level trust and buy-in.

Commitment to Quality

I should have placed this issue first in the list, as I fundamentally think that teams have forgotten their central agile commitment to quality in high TTD environments. Perhaps a strong contributor to this is the general lack of organizational understanding of the dynamics of software quality and software testing.

The commitment to quality doesn't magically appear within every agile team. It needs to be coached and supported within the team by leadership. In this case, I believe it's fostered and developed by a top-down focus. Functional leadership, Scrum Masters, and Product Owners need to understand that they have a responsibility for delivering high quality results. That it's not simply a slogan, but the real truth in the organization. That leadership "means what they say".

And this isn't a simple and narrow view towards quality. No! It's a broad view that focuses on inspections of design and code as a collaborative cornerstone of developing quality products. A mindset of testing as a broad and varied activity – executed by professionals who understand the nuance of functional and non-functional testing. And leadership that also understands the returns that an investment in test automation can deliver – both in test coverage but also in allowing for design and code change nimbleness.

I have a personal test for many organizations –

- Do your developers willingly test when their team needs it?
- Or develop extensive automation hooks as part of the design process?
- Or passionately construct unit and functional automation by pairing with their test colleagues?
- Essentially, do they view all quality practices as an active part of their getting their (coding) job done?

Not by direction or obligation, but because they've bought into the value of quality up-front. If your answer is yes, then you've fostered a whole-team view towards quality commitment. If not... then you've still got some work to do.

Wrapping Up

Chris Sharpe is an active agile coach out of Portland, Oregon. At the 2009 Pacific Northwest Software Quality Conference he presented a paper on Managing Software Debt. In his presentation he referred to TTD as Quality Debt, while more pervasively looking at the debt challenge beyond development-centric technical debt and TTD – instead viewing it systemically within software projects. I mention it because I think Chris has gleaned the true nature of the software debt problem. It's broadly nuanced across all software development activity and an anti-pattern standing between agile teams and their true potential.

Another aspect to this is the serious role that leadership plays in "debt management". I spent several years as a Director of Development in a fairly mature agile shop. I was also their agile evangelist and the sponsor for agile and lean focused continuous improvement.

I had a habit of emphasizing quality and doing things right in almost every conversation I held with the team. I felt that for every ten conversations I had, I needed to emphasize craftsmanship, refactoring, proper testing, and other quality practices in nine of them.

Why?

Because it was incredibly easy for my agile teams to get the wrong impression regarding an effective quality balance, and fall back into their pre-agile bad practices of giving quality lip service.

For example, thinking that time was a more important variable for a project. Or to forget that they were empowered to "stop the line" and fix foundational issues. They had been programmed for years in traditional management and development approaches that rarely served quality the way they should have. I felt that I had to break through these historic habits and patterns with a strong message – shared consistently, clearly, deliberately, and often. I encourage all of you to do the same in your renewed commitment towards recognizing and eradicating Technical Test Debt!

I hope you found these thoughts minimally useful, and I encourage you to "face your debt". Thanks for reading,

Bob.

> biography

Bob Galen
is a VP and Agile Coach at Deutsche Bank Global Technologies in Cary North Carolina. He's also President of RGCG, LLC a technical consulting company focused towards increasing agility and pragmatism within software projects and teams. He has over 25 years of experience as a software developer, tester, project manager and leader. Bob regularly consults, writes and is a popular speaker on a wide variety of software topics. He is also the author of the book Scrum Product Ownership – Balancing Value from the Inside Out. He can be reached at bob@rgalen.com

A personal approach and value-chain thinking – Two pillars of successful testing

by Derk-Jan de Grood

Increasing importance of the personal approach

Whilst working, we desire personal contact. We do not want to feel like we are just a part of an administrative or mechanical super-machine. Recently, new ways of working have taken flight in the corporate world. Not in the first place because these allow for working flexible hours from home, but above all because we like to undertake our work in an engaging and stimulating way. We are looking for satisfaction, and are allowed more and more to act as an individual. As a result of this, a trend is emerging within organizations where personal feelings are no longer a taboo. A personal approach is becoming increasingly important, and in the coming years, more and more organizations will adopt this personal approach.

André is a freelance project manager and recognizes that there is more room for a personal approach nowadays. "As a freelancer I have had a lot of intake interviews with new clients. I realize that more and more often I elaborate how I want to do my work. It is appreciated that I detail both why this way of working fits me, and where my pitfalls are. I enjoy this, because it gives me the feeling that I am being hired because of who I am, and not because I am a machine that can perform a certain task."

Increasingly many professionals learn to show their weaknesses. "Getting Naked" (Lenconi, 2010) shows that companies with a culture where employees can share their doubts and insecurities and openly name their pitfalls can be very successful. Especially in organizations where the human dimension is paramount, testing can have an important role. This is why in future testing will be more and more focused on delivering comfort. In his latest book "Grip on IT" (Grood, 2011), Derk-Jan de Grood points out that IT is traditionally a technology driven sector. For this reason, personal motives are often neglected in daily practice. He investigates how key individuals act in projects that are under high pressure. Project managers and clients are looking for success, comfort and a feeling of "I have got grip on my project". Loss, pain and fear, on the other hand, are strong negative drivers that should not be ignored. What happens if the players are pushed out of their comfort zone and fear of failure is a prominent factor?

Testers will focus more on the personal motives of their stakeholders. By understanding the personal interests and motives of these stakeholders, they can tailor their services.

Yasmin: *"I have met a lot of testers who have regularly had a deja-vu moment. Sighing they explained that it was so difficult to establish a mature testing process, that they got insufficient commitment from their managers, and that they felt that the organization was not really interested in testing. Recognizable, but not necessary. I myself try to understand what drives my clients: what does this project mean for them, when will they start to worry, and what criteria are they themselves being evaluated on? It is these things which determine what decisions will be made. Some project managers will never really put themselves in a vulnerable position, but the norm is shifting from strictly business to more personal. This will increasingly enable us testers to add value. When a client is afraid that the users will dislike a system, then we will come up with an activity to investigate whether this is true. For example, a demo or process simulation. If the customer indicates that he is worried about the performance of the system, then we plan a focused review, or we organize a session with a performance specialist, business architect and developer. In short, we do not test because we are testers, but we add value because we provide an answer to our stakeholders' questions. In cases where we lack techniques, measures and knowledge, we will as testers think out-of-the-box and collaborate. Here, we will increasingly draw on other disciplines, such as Lean, the petrochemical, automotive or the gaming industry."*

By choosing a personal approach and by connecting with the personal experiences of the stakeholders in the project, testing is changing from being methodically driven to being context driven. The emphasis shifts toward the value that the stakeholders attach to the output of the process, and this varies from person to person. Testing becomes a process with staff from different disciplines. Therefore, testers will find new challenges in the near future, as they will have to be more open themselves. When they collaborate, they will have to indicate where their strengths lie, and what their pitfalls are. They will also have to learn how to engage in a personal dialog, so they can build trust and empathize with others. In addition we will have to find substantial new ways to deliver stakeholders that what they are so eagerly looking forward

to: a supplier that helps them reduce their worries, giving them assurance where they are uncertain, and above all increases the success of their projects.

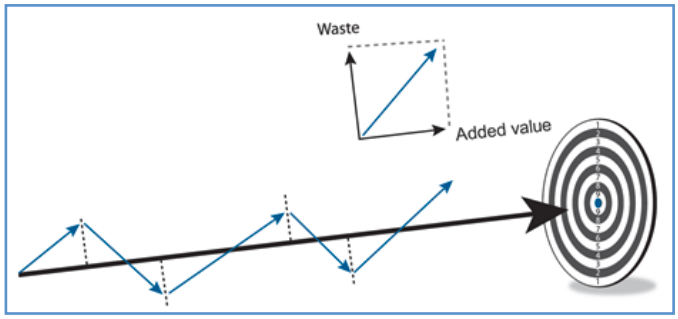
Thinking based on value chains

The personal approach as detailed in the last paragraph helps increase a project's chance of success. By having a good understanding of the needs of the business, the focus shifts away from the activities themselves to the value of output of the process. The second pillar that a project's success relies on is a good organization of the value chain. This requires a good understanding of the business. Once we have this understanding, we can 'align' our activities.

The value chain originated from manufacturing and is known from the theory of constraints. It has also made its mark within Lean. A value chain can be well explained by means of a conveyor belt process. The product on the belt undergoes a number of operations. Every operation adds value: it leads to a shippable product. For every operation, 'waste' can also arise: operations that cost time and materials, which will not end up in the final product. Logically, we want to accentuate adding value in every step of the chain, whilst reducing waste.

Activities contribute to the aim pursued, whilst also having a waste element.

Although the value chain exists since 1985 (Porter, 1985), its importance will strongly increase over the next years in the IT industry as a whole, and in the testing services sector in particular. Due to



market expectations and economic developments, there is high pressure to perform better and more efficiently. Thinking in value chains helps to focus on those developments that matter, aiding in winning the efficiency battle. In the realm of IT, thinking in value chains means that every party involved in software development is aware of their contribution to the final result.

André is employed by a firm working in the energy sector, which has adapted the value chain methodology. *"Within our organization we are strongly aware of the fact that all energy must be put into creating solutions which add value, as we dislike wasting energy in our sector. Therefore, management has put forward some clear principles".*

Each step has a clear goal

Employees can explain the purpose of the step on which they are working. This is expressed in clear acceptance criteria or a clear 'Definition of Done'. For them it is clear which problem must be solved. Most activities are in fact still done on commission. So the

client has made clear what his needs are, when the results meet his or her expectations, and what the (sub)product is being used for.

The success criteria of the next step are taken into account

An activity or process step does not solely have a goal, but also provides something. Usually, this is an input for another discipline. A draft for instance can be used by developers to build the system, by testers as a test basis and by the users to verify if the system is fit-for-purpose. In a value chain, the 'pull' is more important than the 'push'. This results in a lot of attention for subsequent steps. The goal is to proceed through every step of the chain in the most optimal way, and every previous step facilitates this process. This means that employees know who they are working for, and what success criteria apply to the receiving party.

Clear reporting

Parties inform each other. The communication between these parties is clear regarding the chosen approach, and how this approach adds value towards the ultimate goals. If there is waste, the parties can explain why this waste is necessary and thus acceptable. The reports are written in an unambiguous way, so that they answer questions. The parties involved know what they can and have to do with the information presented to them.

Business can detail the value of each chain

Due to the points mentioned in the previous paragraphs, the business is involved. They know which parties are working to achieve their goals. They ask clear questions, and they know which party should answer which question.

These guidelines seem to work well in practice. I regularly use them myself to determine whether my focus is correct.

For testers, the properties mentioned above mean that they have to be able to explain their role and its added value. Testers should be knowledgeable about the possibilities for the test strategy, and the decisions they make for this test strategy should be well informed. The testers should connect with the business in order to identify their needs, allowing them to take the right decisions gaining the business' commitment. Testers will also have to speak the language of developers (who have to solve issues, and who need a clear explanation why certain tests need to be performed), administrators (who will be using the tests during regression testing), users (who will work with the application), etc. Testers will need to switch between the macro and micro levels even more than they are doing now. Whether they rotate in a SCRUM team or have the role of acceptance manager, they will have to be able to identify risks, and be able to clearly explain these risks to the stakeholders, what this means for the next steps, and for the final goal.

References

(Porter, 1985) Michael Porter *Competitive Advantage: Creating and Sustaining Superior Performance.*

(Lenconi, 2010) Getting Naked: A Business Fable About Shedding The Three Fears That Sabotage Client Loyalty. ISBN-13: 978-0787976392

> biography

Derk-Jan de Grood
works for Valori and is an experienced test manager and advises companies on how to set up their test projects. He published several successful books. Currently he is co-authoring a book on the future of testing. With this book, he and his fellow authors aim to help their fellow testers anticipate future needs and demands, and believe that this will help in keeping testers happy in their job and in sustaining the value of the profession. This article is closely related to one of the chapters in the forthcoming book.

Derk-Jan is a regular keynote speaker at national and international conferences and is passionate about his profession. He gives lectures and provides training sessions for fellow professionals in the trade.
Yasmin and André are fictional testers like you and me. They could be sitting in the cubicle next to you.

SWE Guild

The Software Engineering Guild

A Morning Revelation

by David Zisner

It started like any other morning: at around 6:15 am, my daughter Noga (2.5) came into our room and demanded my attention in her own special way – which mainly involves standing on top of me and saying "look at me!" over and over with a huge smile. I cooperated, but when Shahar, our 4-month old baby started to also demand his share of attention, it became clear that I was done with resting. Time to start the day.

At that point, our usual routine started: breakfast for everyone, diapers, getting dressed, braids for Noga, new set of clothes for Shahar (don't ask), shoes, bike – and off we go. There were a few tears involved, none of them mine, thank God, but by 8:17 they were both in their kindergartens and I was on my way back home to start the day which, as it felt to me, was already 5 hours long.

My kids are great, I love them, but let's face it – the morning routine is not always the most creative or mind-boggling thing in the world. It's basically a set of operations that must be performed in a certain order to achieve a desired result without excessive effort. After all – there's a day's work ahead of us.

Since all my professional life revolves around automation, I spend a lot of my time helping companies develop automated tools for validation and production, and so my mind immediately went to finding a solution to a well-known problem: a set of operations needs to be performed daily to achieve a desired result with minimal cost. How do I automate our mornings at home?

Thoughts of Robo-Dad came and went, and while the perfect solution still eludes me, I suddenly realized why some companies fail to get their chips out to market on time and in good quality: Neglecting to invest in automation and choosing instead to manually test and integrate their product, they are actually doing what I do at home every day: they perform the same set of tests daily (or whenever a new version of the product is released), expecting to produce exactly the same coverage and reach the same conclusion. And since the coverage they need to attain increases as their product matures, and as release cycles shorten when approaching production, they actually need to do more with less time, always maintaining the same level of quality. No wonder the development and integration teams are exhausted as production approaches!

Exhaustion leads to errors, errors lead to false conclusions, which in turn lead to customer dissatisfaction which is hazardous to any company, especially to start-ups that are just building their reputation. Also, while your engineers are busy performing manual tests – do they really have the energy and availability to come up with new test cases and methods, or are they just trying to keep up with the releases?

So while it's of little significance whether the breakfast I prepare tastes a little different every day, companies should make the best effort to make sure that their integration and validation environment is consistent, fool-proof and preserves their engineers' energy to operations that actually require their brains and focus – analyzing test results, or writing more tests to increase the automated coverage. Or building Robo-Dad, and blessing all of us with just one more hour of sleep every day.

> biography

David Zisner

My name is David Zisner – an electronics engineer with extensive experience in system design, development and integration.
On top of being hands-on, I managed development and integration teams in both corporate and start-up environments, collaborating with teams both local and international. In 2011 I started LeAndolini Solutions, a service provider in the field of automation for validation and production lines. LeAndolini helps leading chipset and system vendors to design and build the automation tools that enable them to both validate and manufacture their products. In our work we focus on the R&D needs as well as the link between R&D and operations. Apart from my extensive involvement in automation activities, I am an avid swimmer and recently swam my first open-water marathon.

Critical Success Factors for a Test Center of Excellence

by Erik van Veenendaal

The theme of this issue of testing experience makes me think back on an assignment I did a few years ago at an organization that developed highly complex multidisciplinary safety critical systems. It was probably the most successful organization I have been involved in when it comes to setting up a test center of excellence and implementing test improvements. On a recent flight back home, I relaxed and tried to analyze what really made the difference. Many things came to my mind, but there are probably four essential success factors that I want to share with the readers. As always, trying to improve or manage an organization using just a top-down approach doesn't work. A combination of top-down and bottom-up is usually most successful. During my analysis, I came up with two top-down and two bottom-up critical success factors.

Top-down approach

Test policy

A test policy, if well written and with the right level of detail, provides direction on the improvements, states the values of the testing organization, shows how testing adds value to the overall business objectives and defines some key test performance indicators. Not without reason does the TMMi model have "Test Policy and Test Strategy" as one of its main process areas already at maturity level 2. At this company the test policy was established in a way that policies are intended (!), and thus provide added value. It was written on one page only, well distributed and put on walls, e.g., near the coffee machine and copier, to ensure everyone was aware of it. On a yearly basis, once the business policy was re-discussed and agreed, the test policy was re-discussed in a brown paper session with managers and all test managers. Are the values still correct? What are the main problems we encounter? What should we focus on this year? How do we add value to business? These are examples of questions that were discussed in the session, with the result being documented in an updated test policy.

Leadership

Whenever I have been really successful in my career, there has almost always been a manager that made the difference. I recall many names of managers that immediately come to mind when thinking of successful projects. Somehow it is sad that success should depend on one person only, but on the other hand real leadership makes the difference. Leadership has been described as the "process of social influence in which one person can enlist the aid and support of others in the accomplishment of a common task". Also in this organization, I encountered a real leader. Someone with a strong personality, who was there for his people when things got rough, who was able to motivate them when needed, who had a vision toward the future of the company and the role of testing, but also, and probably above all, a pleasant, honest human being with integrity. Leadership is much more than providing resources. Real leaders just make the difference, find them!

Bottom-up approach

The test managers' meeting

One of the things I learned from Watts Humphrey many years ago is that critical stakeholders during change management are those directly responsible for the project, e.g., the project manager and test manager. They tend to be directly confronted with delays due to new and innovative processes, while their job is to deliver a quality product on time. Thus they are usually not too enthusiastic when it comes to process improvements to their projects. At this company, although a person was assigned as a process improvement facilitator, the test managers' meeting was the main driver of any improvement. At their weekly meeting they would discuss, in addition to (test) project issues, improvement ideas, the status of improvement actions, etc. They were the ones to decide whether an improvement had added value or not, making it very practical, and also with a strong focus on added value in projects using the test policy as their reference framework. When a decision was made to define and implement an improvement action, a volunteer was sought amongst the test managers that would drive the specific action. Note that the improvement actions were not process oriented only, but also concerned tasks like seting up and organizing a training session, tool selection, developing a supporting Excel template, etc. As a result, little or no change management actions were required for the test manager, a typical difficult stakeholder. They were now in charge of doing things they perceived had added value to the projects, either short or long term.

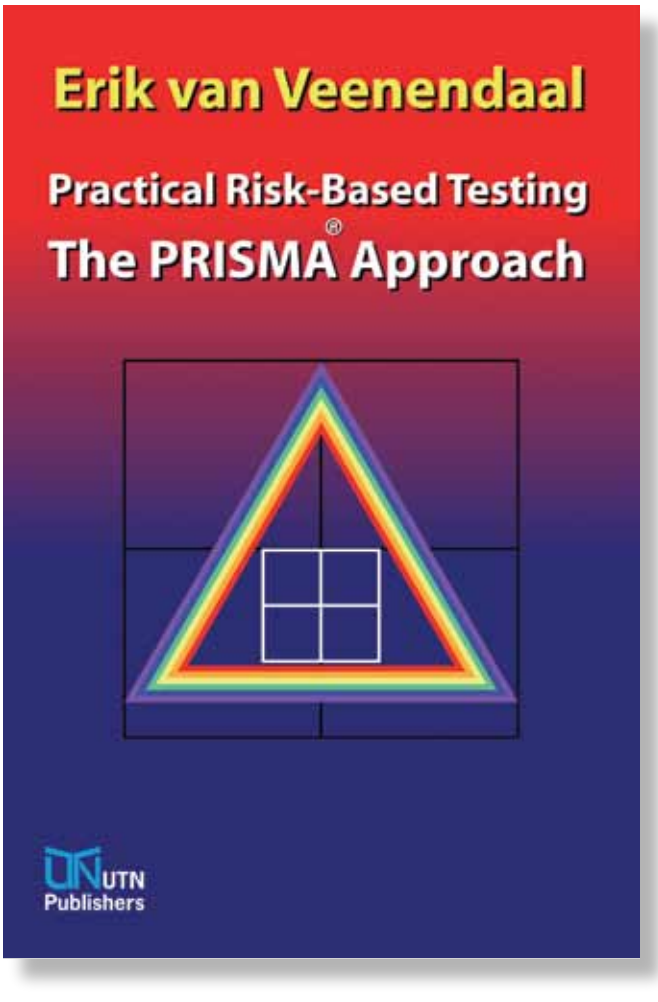
Everyone involved

Although the test managers were driving the improvements, they would not necessarily do it all themselves. Normally they would take the action back to their team, discuss it and make it a team assignment. In this way all testers became involved in building a high quality test organization and test improvement. There was almost no need for change management, it was already their test process. Of course implementation could still be difficult for many reasons, but at least there was little to no resistance and people had a positive mind-set towards the changes. Important in this context was also a monthly test knowledge sharing meeting where testers and test managers would present their personal experiences regarding a new template, test design technique, using a tool, etc. Again the test improvement coordinator was acting as a facilitator rather than a manager. A test process was being built that was already everyone's process.

You may notice that the critical success factors described above are strongly related to change management and people issues. I believe people do make the difference. A strive to become a testing center of excellence using just a process orientation will fail sooner or later. Real leadership, a clear policy, mission and direction, and a focus on change management and people issues can make it happen. But remember, changes do take time!

> biography

Erik van Veenendaal *(www.erikvanveenendaal.nl) is a leading international consultant and trainer, and a widely recognized expert in the area of software testing and quality management with over 20 years of practical testing experience. He is the founder of Improve Quality Services BV (www.improveqs.nl). He holds the EuroSTAR record, winning the best tutorial award three times! In 2007 he received the European Testing Excellence Award for his contribution to the testing profession over the years. He has been working as a test manager and consultant in various domains for more than 20 years. He has written numerous papers and a number of books, including "The Testing Practitioner", "ISTQB Foundations of Software Testing" and "The Little TMMi". Erik is also a former part-time senior lecturer at the Eindhoven University of Technology, vice-president of the International Software Testing Qualifications Board (2005–2009) and currently vice chair of the TMMi Foundation.*



NEW PUBLICATION

PRISMA is an approach for identifying the areas that are most important to test, i.e., identifying the areas that have the highest level of business and/or technical risk. The PRISMA Approach provides a practical, realistic and effective way to introduce and maintain a testing approach that manages product risk, provides guidance to the testing effort and supports ongoing process improvement efforts. Risk identification, risk analysis, risk mitigation and risk monitoring (with a special focus on reporting) are explored and presented in this book.

Erik van Veenendaal is a widely-recognized expert in software testing and quality management, a leading international consultant and trainer with over 20 years of practical experiences, and the founder of Improve Quality Services BV.

ISBN 9789490986070
pages: 136
price € 19.90.
Order at www.utn.nl

"As the title indicates, this is a practical and straightforward approach to risk-based testing that addresses the real world needs to efficiently apply a testing methodology that satisfies the needs of the stakeholders, creates an understandable road map and allows the testers to apply their unique knowledge and experience. The book is easy to read, provides clear examples and leaves you excited to apply what you've learned."

Judy McKay

The Test Center of Excellence

by Mariya Vankovych

Very often we use the terms "quality", "goodness", "excellence", etc., but what do those terms really mean? We can find the following definitions of these terms in Merrian-Webster dictionary: "quality" – superiority of kind; degree or grade of excellence; "good" – adequate, satisfactory; "excellence" – is the quality of being excellent; an excellent or valuable quality.

Generalizing, "excellence" means being of outstanding quality or remarkably good. However, who are the quality and goodness supposed to be for: for me, for you, for some third person? The definitions mention "adequate" or "valuable quality", meaning that the item in question, in our case software, should be useful. This, however, also depends on the end user – one program that fulfills my needs may well be useful for me, but not my colleague that has other needs. From his/her point of view, the same program may be useless and have no quality. Everyone has different needs. Therefore any product receives its characteristics of quality and excellence only in the context of the end user.

(See Fig. 1)

There are many classifications of quality. For example, Isabel Evans in her book "*Achieving Software Quality through Teamwork*" classifies quality as user-based; value-based; manufacturing; product-based; and transcendent. Based on this classification we can choose the best way for achieving "excellence" in the product development.

Recently more and more people say that the world changes so rapidly that there is no time for testing and verification. We need to be very flexible to keep up with changing customer needs. Rapid change of requirements and need for flexibility dictate the way how the software development process is conducted. If company wants to stay on the market, it needs to accept these rules of the game. However, the industry for which the software is being developed also makes its own adjustments to the software development process. In case faulty software can cause damage to human lives, human health, personal data loss, money, customer

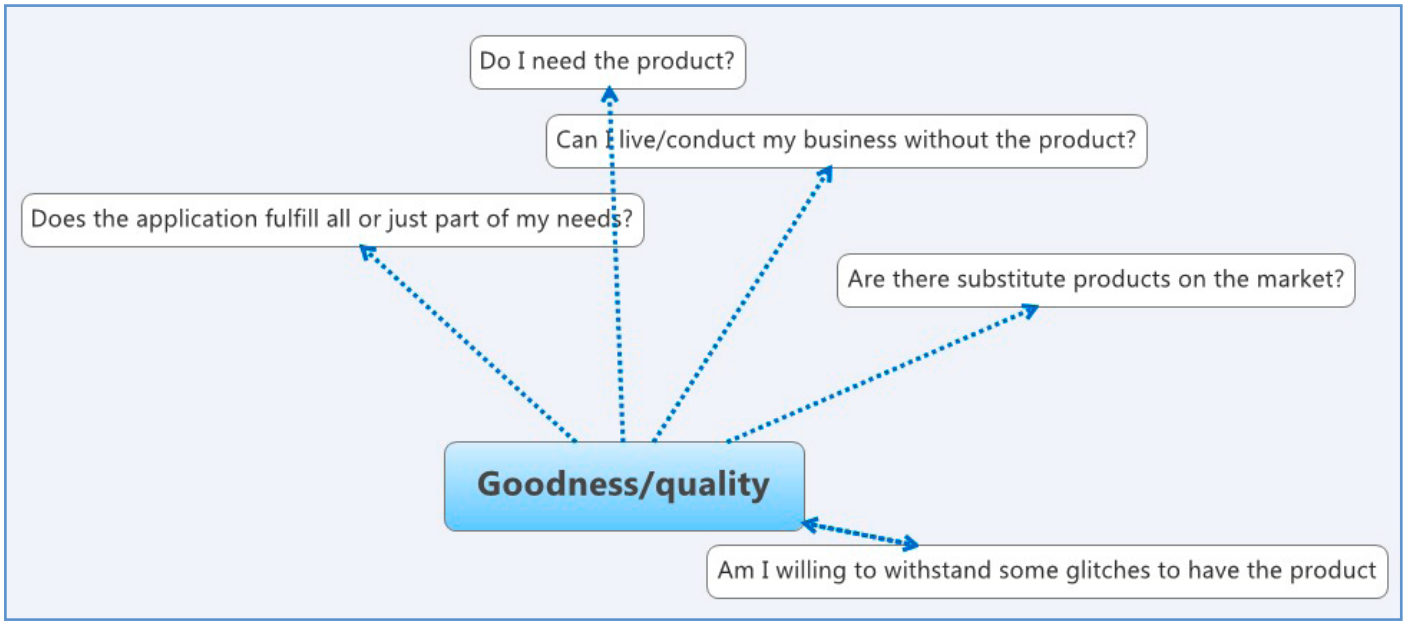


Fig. 1 – Some questions that need to be answered to figure out whether a product will bring quality to us.

Do you want to write an article for the next "Testing Experience"?

If you want to participate in the next issue please follow the steps:

- Download the MSWord template and complete it including a short biography of the author.
- Submit your finished article.
- Our editorial board rates your article.
- Our editor José Díaz accepts or rejects your article.
- If your article is accepted, you send the figures and pictures to be included in the article in the highest resolution you can provide (72 dpi for screenshots, at least 300 dpi for all other image files) and the photo(s) of the author(s) to editorial@testingexperience.com.
- Download and sign the consent form, scan it and send it to editorial@testingexperience.com.
- Our lecturer reviews your article (grammar and spelling).
- You get your article back with the marked changes.
- You accept or reject the changes.
- When the magazine is published, you will receive your article separately in PDF format.

Note

- Please take care of copyrights and registered trademarks.
- We do not accept sales pitches, where a product or company is being advertised.
- Your article must not have been published before.
- There will be no remuneration.

For any questions please write to editorial@testingexperience.com.

Testing Experience calendar

Issue	Article	Topic
September 2012	August 1st	Mobile App Testing
December 2012	November 1st	Testing in the Cloud

www.testingexperience.com

Eclipse Testing Day

Darmstadt, 5th September 2012

Topics

- Testing Eclipse applications
- Testing within the Eclipse Ecosystem
- Testing on Eclipse Projects
- Design for testability in Eclipse
- Case studies of testing projects
- Eclipse tooling and technology for the test process
- Testing as a part of the application lifecycle
- Continuous integration and testing for Eclipse applications

For more information see http://wiki.eclipse.org/Eclipse_Testing_Day_2012

credibility etc., moving forward without thorough validation of the system will definitely not be the desired process.

In an ideal world everything would be excellent, but unfortunately there are some obstacles on the way to achieve this that are hard to overcome in real life. I will describe several of them.

1. How can we guarantee that we know all the dependencies in big projects (analysis and design)? It can happen that we are missing some requirements, or we are not taking into the account some interdependencies of the big system. To achieve excellence in analysis and design the attention should be paid to good documentation that can track changes in requirements, changes in technologies, and be good resources of knowledge transfer in case the domain expert is unavailable. But even availability of documents does not guarantee that everything was analyzed and nothing was missed. Keeping up-to-date documents is time consuming and requires great efforts from employees. Companies achieve excellence in this area by implementing strict policies and by educating employees why this process should be implemented. This approach is good for e.g. medical related industries. Availability of precise and finalized documents makes it possible to guarantee some pre-defined level of acceptable quality, so, in other words, excellence of the software. On the other hand, in case of game web portal or social-interaction webpages no one will opt for this approach as the cost of this effort will be too big comparing to risks that the system fault can produce.
2. How can we guarantee that we know where we are (management)? For sure there are measurements methodologies and reports. Those managers who are working with small, transparent teams and combine managerial tasks with the technical ones, may say that they do not need to measure anything. They see how the work progresses and they see how the team works through everyday personal interaction with the team. Their approach can work depending on industry they are working for and on how many people they have in the teams. At some point it's already not so easy to track everyone's progress with naked eye only. So, how can we make sure that we know where we are, that we know what requirements were covered, and who needs assistance with the assigned task and how the project looks in general? As soon as the first sings of uncontrolled process appear (raise in defects number, no defects raised at all, delays in completing tasks etc.) managers should start implementing measurement process. What's more, the project should be analyzed against risks and estimations before even entering the development phase. Having risks and estimations prepared long before the real work starts, we can be sure that in case of adverse circumstances we still will be able to complete the tasks at time. Considering human factor in the development process, without some kind of measurement and reporting there is a big risk of the "hot-potato throw effect" in the event of post-release issues. In the end it doesn't matter who's fault it was, as the whole team should work together to achieve "excellence", but in reality everyone will be looking for scapegoat. Reports are needed not to know whom to punish, but to understand where the weak points of the team were and how to improve that area in future. So again we have arrived to the point that shows that every time new project starts it depends on the team, on requirements, on team maturity and on the type of measurements and estimation approaches that should be used to achieve excellence in the end result.
3. How can we guarantee that constantly changing requirements will allow us to produce the end product that meets the required quality (management, design, development, validation)? Due to the realities of modern world we can never be sure that what was decided today will be also valid tomorrow. We are constantly ending up with new ideas and requirements in the middle of a project when coding has already started. Should we say "No" to the customer and deliver only what was initially requested? In this case we might deliver something that has no value for the end user anymore, and we could lose both the customer and money. Should we accept all new changes to the requirements? In this case we can end up with an unstable product, or we prolong the SDLC of the product and again we could lose the customer. Of course saying "no" to the customer is not the right option that will be accepted by higher management. Customers pay money so their wishes should be fulfilled. Taking into account the variety of industries for which the software can be produced, it again depends on whether it is the best decision to accept the new requirement but postpone it to the future release, or accept new requirements on the fly. The more severe consequences of the system fault are, the stricter accepting any new requirement we should be.

Generally, there are three main actors in the Software Development Lifecycle: customer, company (upper management), and technical people (architecture, coding, testing) – not taking into account third parties (providers, logistics). All three of them have different needs and goals.

Customer	Company (management)	Company (development)
To pay less	Gain more – spend less	To earn good salaries
To receive desired functionality	To fulfill customers' needs	To have enough time for work
To receive the tool on time	To release everything on time	To release without faults, bugs, glitches
To be able to change requirements at any time	To release without issues	To have stable requirements
To receive the tool without faults, bugs, glitches		To have deadline moved in case of requirements changes
		To have deadlines and work load that are based on analysis, and not fixed from above

As you can see, all of them have one mutual goal – to have an end product that is stable and does not fail – so in some way all of them want an excellent product. But this is only one goal. Others in some way contradict to each other. Customer would like to pay as little as possible for the product. Company would like to receive more income and pay less to employees. Employees would like to receive high salaries. And this is only one of the examples of contradictory interests.

As the needs and goals of SDLC actors are different, the only possibility to achieve excellence for everyone is to find the consensus. Any way to consensus leads through discussion and the results will depend on:

- What do customers need?
- What are customers willing to tolerate to receive what they need(delays, level of acceptable minor issue detection after release, higher cost than expected and so on)?
- Industry and the worst scenarios of faulty system
- Company strategic plans and possibilities

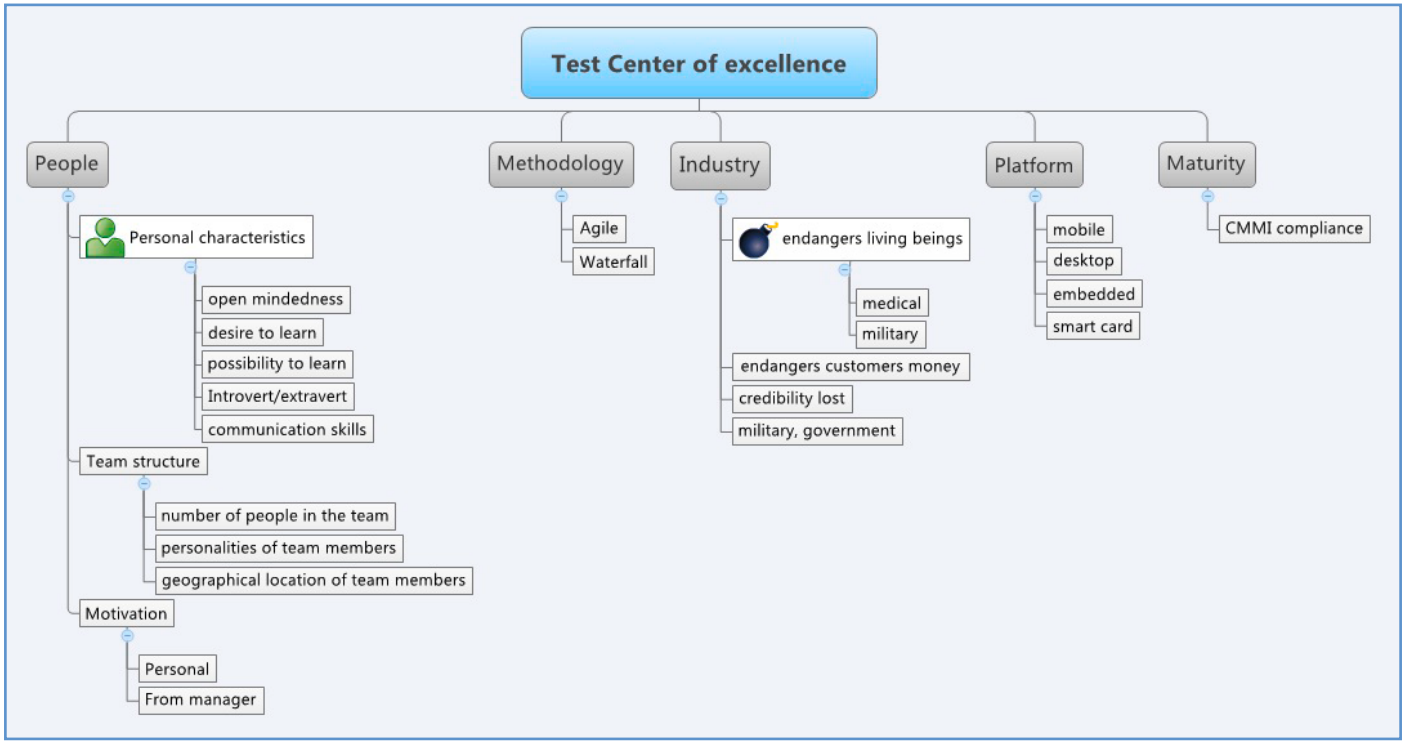


Fig. 2 IT project excellence components

- Employees needs and possibilities
- Methodologies involved
- Platforms involved
- And other factors

(See Fig. 2)

So let's look more closely at the components of test center of excellence. One of the most important components is people. What characteristics should employees have to be able to contribute to excellence? First of all, he/she should be motivated to do good work and it can be achieved by both inner personal motivation (feeling of success, feeling that he/she is part of a big team that is doing very important stuff), or outer motivation (appraisals, bonuses). Employees should have a desire to learn new technologies, and also be given the possibility to do this. What's more, employees should be able to communicate well their ideas: they should be open for criticism, they should be able to communicate issues without assaulting others, and they should be diplomatic.

Another main component of excellence is methodology to use. A decision needs to be taken whether to use Waterfall or Agile? This part is mainly dependent on the industry we are working in. If the program faults can lead to health damage, – death, or serious consequences, we should select the methodology that will allow us to employ good metrics and gain control over the entire SDLC. In this particular case, compliance with CMMI levels will probably be required by customers.

(See Fig. 3)

Excellence also depends on which customers we are aiming at. Is the product going to be a COTS product intended for a wide range of customers, or a tailor-made addition to a COTS product for a specific customer, or a tool made from scratch for a particular customer? In the first case, it's hard to achieve excellence as everyone's needs are different. There will be some overlapping of needs and in the functionality that product offers, but most of the time there will still be areas of customer needs that are not

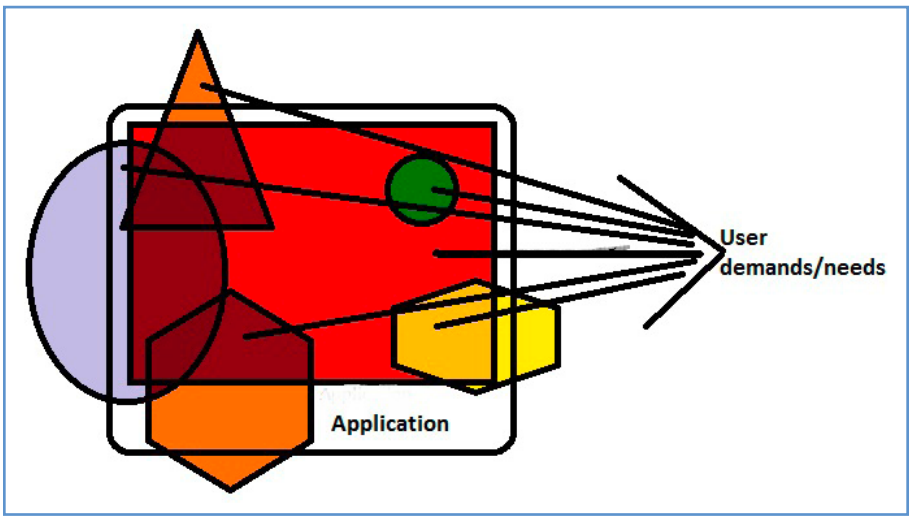


Fig.3 – Product functionality vs. customers' needs correlation

Knowledge Transfer – The Trainer Excellence Guild

Díaz Hilterscheid

Integrating Agile with a Structured Project Culture

by Paul Gerrard & Susan Windsor

Is Agile proving hard to implement in your organisation?

- Have you been successful with agile development but still having difficulty scaling to larger projects?
- Is your organisation unable to release product owners to your agile teams when you need them?
- Do your suppliers or customers have a different interpretation of agile to you?
- Do you need to resource a mixed portfolio of development approaches from a single resource pool?
- Are you lean enough?

» more at
www.testingexperience.com

Date	Place	Price
June 21–22, 2012	Berlin	€ 1,250 + VAT

Testing Mobile Applications

by Jonathan Kohl

Smartphones and tablets are here to stay. Test teams are facing pressure to deliver testing quickly and successfully on mobile devices. When faced with a mobile testing project, many testers find it tempting to apply the same methods and techniques used for desktop applications. Although some of these concepts transfer directly, testing mobile applications presents its own special challenges, which unaddressed means you miss critical bugs.

» more at
www.testingexperience.com

Date	Place	Price
August 30–31, 2012	Düsseldorf	€ 1,250 + VAT

TDD .NET

by Lior Friedman

This three-day program provides a highly interactive exploration of unit test automation, principles and practices. Companies such as Microsoft, Google, and IBM have already realized the potential that lies in Unit Test automation. Following these practices reduces the amount of defects in your software decreasing the effort involved in product development and result in more satisfied customers. Learn how to use frameworks for test writing and how to isolate your code in an efficient and easy manner.

» more at
www.testingexperience.com

Date	Place	Price
Sept 10–12, 2012	Berlin	€ 1,450 + VAT

Specification by Example: from user stories to acceptance tests

by Gojko Adzic

The winner of the MIATPP award at the Agile Testing Days 2011 presents: This two-day workshop teaches you how to apply emerging practices for managing requirements, specifications and tests in agile and lean processes to bridge the communication gap between stakeholders and implementation teams, build quality into software from the start, design, develop and deliver systems fit for purpose.

» more at
www.testingexperience.com

Date	Place	Price
Sept 17–18, 2012	Berlin	€ 1,200 + VAT

covered unless some add-on is developed for particular customers. Another consideration here is about the industry the customers are from. As it was already described before, for some industries such as medicine, government, transport it is vital to aim at zero-defect products as the cost of a single defect can be too high.

At present almost each day new innovative gadgets are created. That constantly adds new platforms for verification and those platforms need different verification approach, different technologies, and even different design of verification process. Nowadays almost all software products that aim at broad mass of society have their mobile versions. And taking into account how many mobile devices are available on the market today, it's really impossible to verify all software products on all devices. Thorough analysis of the market should be done to find the golden middle of what should be tested, on which platform it should be tested, and when we should stop verification process to achieve the excellence of the product for all actors of the SDLC.

And the last but not the least, the level of maturity of the software development company has a strong hold over the excellence of the produced software. The higher the maturity (processes and polices are established and the company employees are already used to all the process) the more excellent the end product is.

Excellence is the art of keeping balance, the art of finding the correct proportion of what is needed, how it can be achieved, and what effort should be put to achieve that need. The worst pitfall is to concentrate only on one facet of the SDLC. Seeing only one side of the process, or taking into account only needs of one actor of the SDLC will lead to prolongation of production process, making software too expensive, exhaustion of development team, and to unfulfilled needs.

Speak, communicate, debate, plan, analyze and consensus will be gained and through it will come excellence.

> biography

Mariya Vankovych
holds a Master degree in Applied Linguistics but decided to start her software testing tenure over 6 years ago at N-ix ltd., Ukraine. Currently she is working at Parexel International and is specializing in web testing.
She actively follows others testers' blogs and tweets and can be followed on her own blog http://quality-and-life.com/.

The DNA of a Highly Effective Testing Center of Excellence (TCOE)

by Nishant Pandey

The past few years have witnessed increased competition and financial pressures on IT departments. This has led to increased understanding and appreciation for specialized skills in various domains including Software Testing, Project Management, Business Process Management and Business Analysis. Organization's structures have subsequently evolved to enable them to focus on building capabilities in specific areas of expertise. This focus and commitment towards developing excellence in a particular capability can often use establishing a Center of Excellence (COE) as a catalytic force.

A Center of Excellence (COE) can be defined as an organization of people, processes, methodologies, tools and infrastructure that aims towards highest standards of achievement in a particular sphere of expertise. Setting up a COE can also help elevate the significance of the particular product or service that is provided by an organization.

This article focuses on the aspects that could help make a Testing Center of Excellence (TCOE) highly effective and fit for the purpose for which it was established.

Testing Center of Excellence (TCOE) basics

A Testing Center of Excellence (TCOE) can be defined as an organization of resources (people, processes, methodologies, tools and infrastructure) aimed towards helping the organization achieve excellence in the software testing, with specific focus on the current and future needs of the organization. Setting up a TCOE could prove to be a very wise investment. It is a good practice to create a charter that describes the goals, stakeholders, priorities, success criteria and financial considerations for the organization's Testing Center of Excellence (TCOE).

The reasons for creating a TCOE could be varied depending upon various factors. The most common factors that impact the structure, scope and intended role of the TCOE include:

- Nature of business
- Organization size
- Quality policy
- Business model for IT needs
- Vision and mission of organization

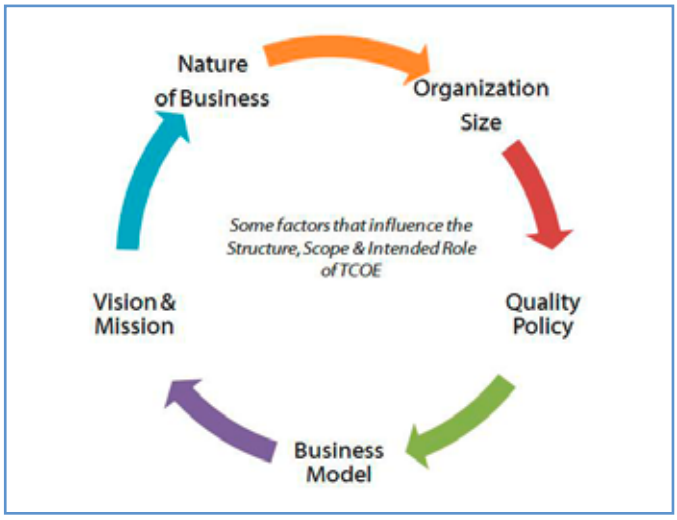


Fig. 1: Factors that impact Structure, Scope & Intended Role of TCOE

A TCOE that is a part of the technology department of a leading bank would have different focus and priorities than a TCOE that is a part of an IT service provider specializing in providing testing services to clients in various domains. Or, for that matter a product company which aims to limit its IT expenditure and looks at testing as primarily a way to deliver the product bug-free, is inclined to have different business objectives and thus a different expectation from its TCOE.

Effectiveness of a Testing Center of Excellence (TCOE)

The aims of a highly effective TCOE need to be consistent with the unique requirements and capabilities of the organization. The effectiveness of a Testing Center of Excellence (TCOE) depends on whether it is able to play a catalytic role in supporting the various departments and stakeholders that depend upon and support the TCOE.

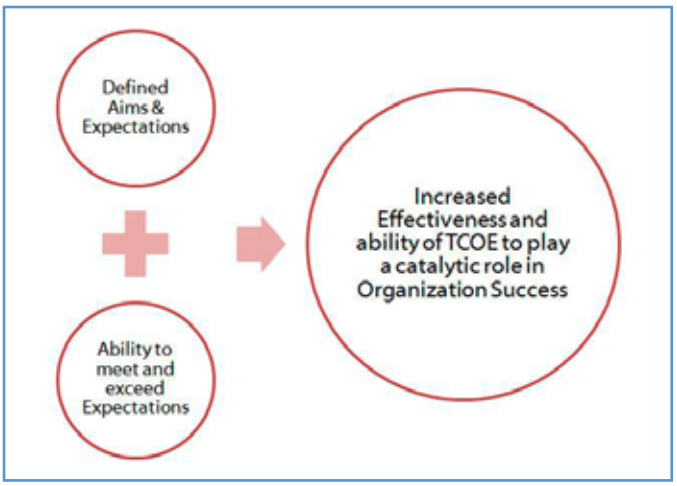


Fig. 2: Factors impacting effectiveness of a TCOE

Frequently, organizations seek to achieve all or some of the following needs by investing time, money and effort in setting up and institutionalizing a TCOE.

- Provide focus and direction to organization's testing efforts
- Improve and sustain current testing capabilities
- Build capacity to handle peaks and valleys concerning the volume of testing work
- Enhance testing domain skills specific to the organization's needs
- Gain insight into industry best-practices, trends and tools
- Facilitate knowledge sharing with other departments and teams
- Improve delivery of testing services
- Facilitate exchange of ideas and dialog within the organization
- Develop the ability to respond quickly to needs of internal stakeholders
- Develop the ability to respond quickly to external stakeholders (RFIs / RFPs)
- Ensure representation of interests of testing teams within and outside the organization
- Optimize the internal business process related to deployment of testing resources
- Enhance customer satisfaction
- Anticipate customer issues and problems
- Proactively lead the creation of an environment for empowerment, agility, and learning
- Facilitate regular review and benchmarking of testing organization's performance
- Participate actively in interactions with leading thinkers in both academia and industry
- Facilitate career growth of employees in the testing organization
- Create a strong brand and reputation, both internally and externally

The above list is not exhaustive and is representative in nature.

Building blocks of an effective TCOE

While there might be slight differences in priorities of various TCOEs, the following parameters represent the key building blocks and focus areas of a highly effective TCOE.

- Delivery of testing services
- Support for customers and internal stakeholders
- Thought leadership, learning and innovation
- Career development and employee retention
- Continuous improvement of products and services

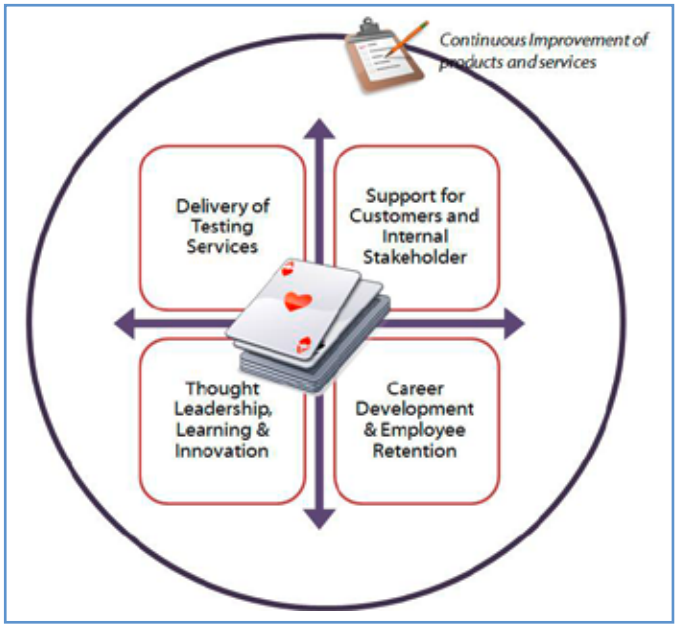


Fig. 3: Building blocks of an effective TCOE

It might be useful to view the activities of a TCOE as elements of one of the building blocks listed above. The sections below explain the individual building blocks in greater detail.

Delivery of testing services

Delivery of testing services is the most basic of the features supported by the TCOE. The TCOE can help enhance the ability of the testing organization and enable it to deliver testing services that are timely, cost effective and of the desired quality standard to its internal and external clients.

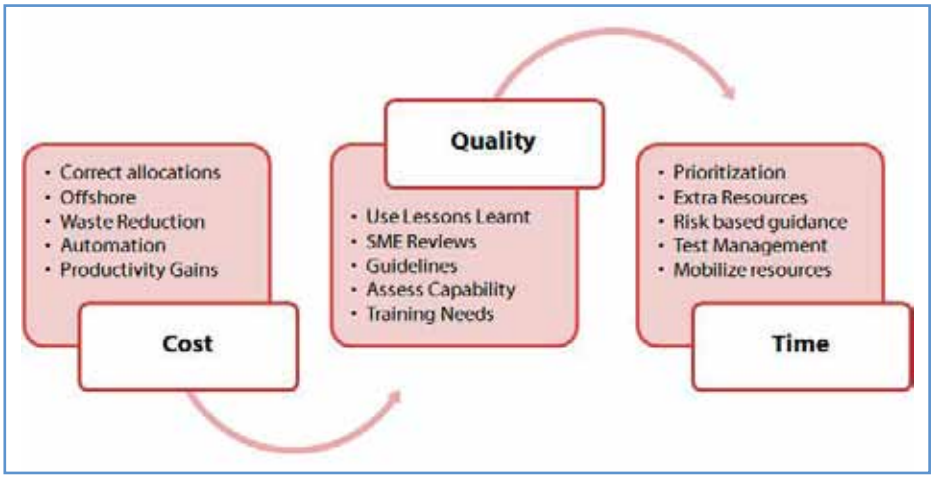


Fig. 4: Pillars of delivery of testing services

Senior Consultants IT Management & Quality Services (m/w) für Agile Softwareentwicklung und Agilen Softwaretest

Wachsen Sie mit uns – wir stellen ein!

Zur Verstärkung unseres Teams IT Management & Quality Services suchen wir engagierte und qualifizierte Kolleginnen und Kollegen.

Wir sind eine marktorientierte und innovative Berliner Unternehmensberatung mit 40 Mitarbeiterinnen und Mitarbeitern. Neben unseren hochwertigen Beratungsleistungen in den Bereichen Financial Services, IT Management & Quality Services, Training Services und IT Law schätzen unsere Kunden ebenso die durch den Unternehmensbereich Events & Media verantworteten internationalen Konferenzen und Publikationen rund um die IT-Qualität.

Profitieren Sie von unserer Erfahrung. Wir bieten

- interessante und anspruchsvolle IT-Projekte in den Bereichen IT-Prozesse und Projekte, MA-Qualifikation und Coaching, Softwaretest und Testmanagement sowie Architektur und Sicherheit
- eine direkte Zusammenarbeit mit der Bereichsleitung IT Management & Quality Services sowie den Teamleitern
- die Weiterentwicklung in einem flexiblen Unternehmen

Sie suchen das Besondere? Wir auch!

Lassen Sie sich anstecken von der kollegialen Arbeitsatmosphäre in einem starken und motivierten Team.

Sie haben

- eine fundierte Ausbildung oder ein Studium absolviert (z. B. Informatik, BWL, Mathematik) sowie mehrjährige Berufserfahrung als IT-Consultant in einem Fachbereich bzw. in der Organisation eines Unternehmens oder bei einer Beratung in entsprechenden Projekten
- Erfahrung mit agilen Entwicklungsmodellen und -methoden sowie mit in diesem Umfeld bewährten Testverfahren und Tools
- Erfahrung mit Change-Prozessen von einer traditionellen Entwicklung hin zu Agilen Methoden
- Erfahrung als ScrumMaster, Product Owner oder in ähnlicher Funktion
- Kenntnisse in der praktischen Anwendung von Methoden und Standards, wie CMMI®, SPICE, ITIL®, TPI®, TMMI®, IEEE, ISO 9126
- Vertriebserfahrung und Sie erkennen innovative Vertriebsansätze

Sie verfügen über

- Eigeninitiative und repräsentatives Auftreten
- eine hohe Reisebereitschaft
- gute Englischkenntnisse in Wort und Schrift

Bewerbung

Bitte senden Sie Ihre vollständigen und aussagekräftigen Bewerbungsunterlagen einschließlich Ihrer Gehaltsvorstellungen ausschließlich per E-Mail an:

hr@diazhilterscheid.de

Díaz & Hilterscheid Unternehmensberatung GmbH
Bereich Personal
Kurfürstendamm 179
10707 Berlin

Telefon: +49 (0)30 74 76 28-0
Telefax: +49 (0)30 74 76 28-99
E-Mail: info@diazhilterscheid.de
Webseite: www.diazhilterscheid.de

As a part of this function, the TCOE could focus on activities like (but not limited to):

- Building a highly professional and well organized testing capability
- Improving delivery, project support and automation of testing services
- Providing guidance to test teams with regards to testing methodology, standards and approach
- Planning capacity in order to meet current and future delivery needs
- Providing good governance for the testing stream
- Ensuring that expectations are articulated and communicated to testing delivery teams

Support for customers and internal stakeholders

The TCOE should be able to assist and support the various departments that have been identified as stakeholders. These could include project teams of current projects and initiatives, marketing and sales teams, product management teams, Business Analysis COE, and Project Management COE. An effective TCOE can help in establishing channels of communication with internal and external stakeholders for an organization. A culture of sharing knowledge and leveraging best practices across various COEs can be of tremendous help.

Fig. 5: TCOE and its interfaces (internal & external)

In IT organizations that provide testing services to their customers, this arena is very important and usually requires a great deal of effort and attention. Time is of the essence while responding to important RFIs (Requests for Information) or RFPs (Requests for Proposal).

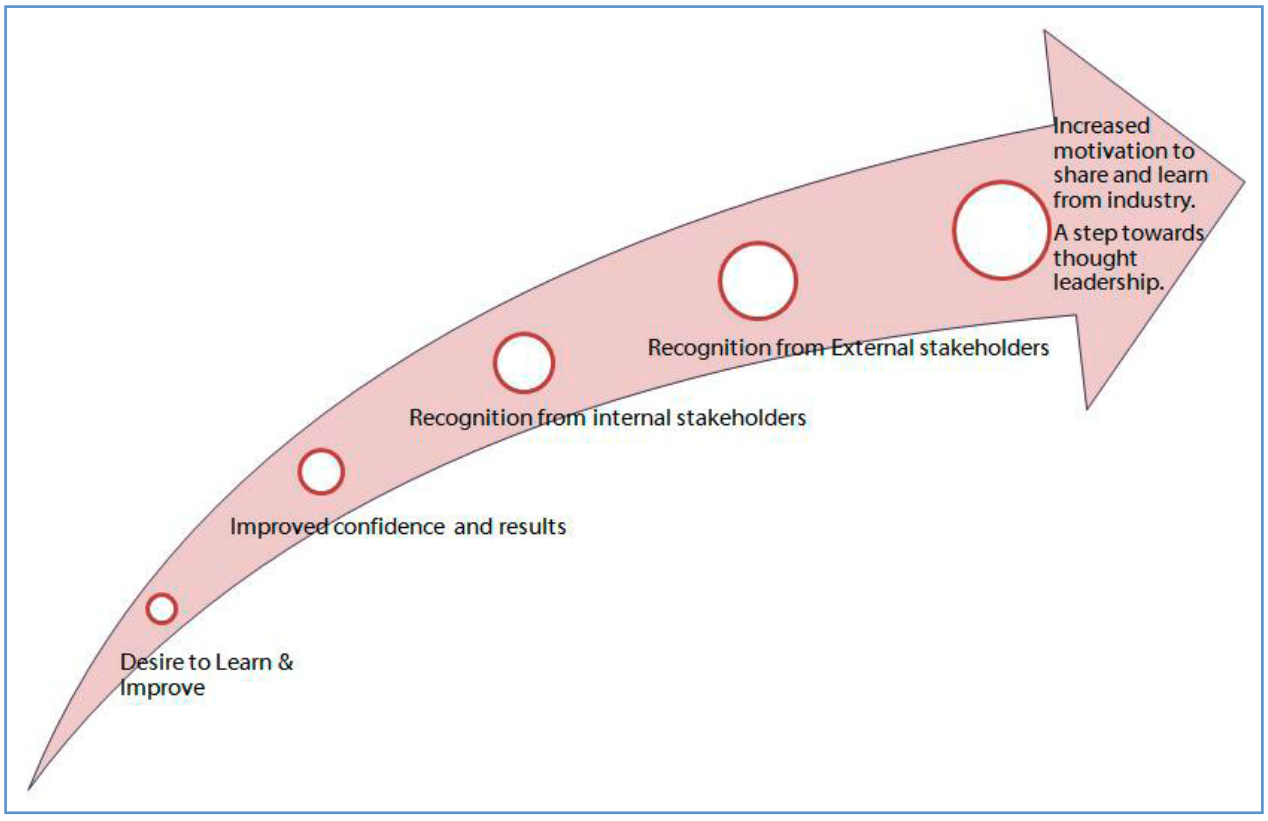


Fig. 6: Path towards thought leadership.

A highly effective TCOE can help collaborate with the marketing and sales professionals to prepare these responses in a timely fashion. Having the testing expertise organized in the form of a TCOE also helps ensure correct prioritization, accurate resource allocation and precise organization focus; improving delivery of testing services.

Thought leadership, learning and innovation

In order to ensure that the latest skills with regards to tools, technologies and processes are utilized in an effective TCOE, it is important that a conscious effort be made towards learning and innovation. A TCOE can help a testing organization stay ahead of its competition or stay abreast with the latest trends in testing with respect to tools, processes, concepts and methodologies.

A culture of innovation can spearhead the creation of new forms of advantages that help realize the true potential of testing and the TCOE. A highly effective TCOE that is successful in establishing a culture of innovation can help an organization establish itself as a thought leader.

It is the recognition for innovative ideas and demonstrated confidence in sharing these new insights and ideas that can help establish the TCOE and its members as thought leaders.

Career development and employee retention

An organization's success is dependent on its ability to train and retain the dedicated and bright employees. This is a parameter that is especially true for knowledge workers. A TCOE can help provide focus and direction to career testers and other employees whose daily work involves interaction with software testing (like business analysts, developers and project managers).

A TCOE can provide support in terms of guidance, access to metrics and data from past projects, best practices, training on tools and processes, and access to established Subject Matter Experts (SMEs). A TCOE can help significantly increase job satisfaction and chances of career success by ensuring access to mentors and trainers who have demonstrated skills in the field of testing or a supporting capability. From the career development perspective, it is important that the TCOE's training plan is holistic, encompassing testing skills, domain knowledge, application knowledge, and general IT and management skills.

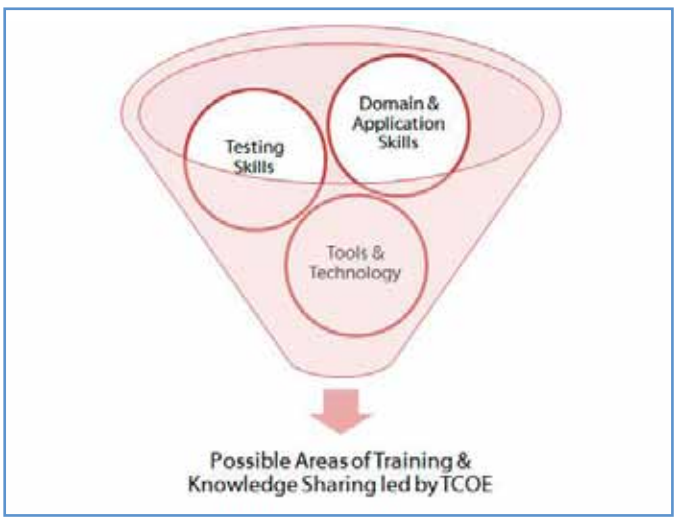


Fig. 7: TCOE and culture of learning.

The investment made in establishing a culture of learning and development helps in many ways including enhancing employee satisfaction and developing the possibility of retaining the best and the brightest.

The TCOE can help serve as the hub for sharing of new and beneficial knowledge and contacts, building relationships at work, getting to know better ways to solve existing problems and learning from each other's experience.

Continuous improvement of products and services

An effective TCOE should relentlessly pursue innovation, creativity and incremental improvements. The focus on eliminating waste, improving processes and products, increasing testing productivity and adding value for the various stakeholders should be one of the core agendas of the TCOE.

Initiating a program to benchmark metrics for continuously assessing the progress and providing feedback for corrective actions is recommended. TCOEs can maintain repositories of lessons learnt from various testing engagements and ensure that these are considered to reduce risks in future projects.

Some organizations that have mature TCOEs use established appraisal, assessment and audit methods to gauge the maturity of their TCOE and the organization's testing processes.

Common pitfalls

A highly effective TCOE is one that has an optimized operational framework for delivering testing services, optimum capacity, and the required capability to accomplish testing related objectives in an outstanding manner. This section deals with some problems that could reduce the advantages of a TCOE and must be handled appropriately to ensure successful returns from the TCOE institutionalization.

Below is a list of common pitfalls:

- Lack of management support
- Lack of clear roles and responsibilities
- Under-estimating time and effort for TCOE setup
- Trying to implement too much at once
- Lack of communication between various stakeholders
- Inability to understand, document and articulate expectations from TCOE
- Difficulties in stakeholder identification
- Inappropriate mapping of roles with team members

One way to avoid these common pitfalls is to ensure that the sponsorship, scope, direction, focus, terms of engagement, processes, roles and responsibilities are agreed and approved in the early stages of implementation and institutionalization of TCOE.

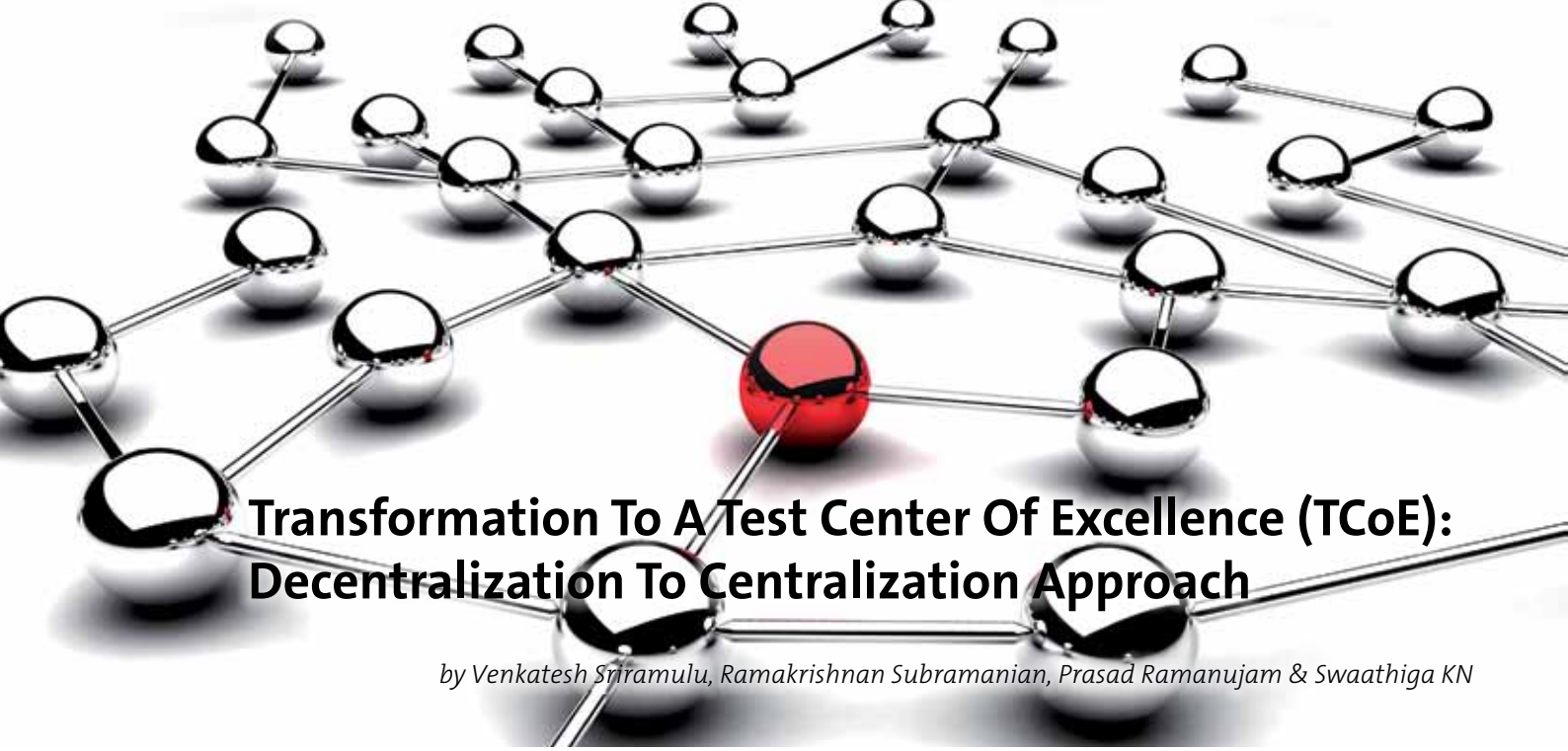
Conclusion

A TCOE is a strategic tool that an organization can leverage for deriving sustained and maximum benefits from the practice of software testing. The returns can differ based on the needs of the organization and the approach it takes towards implementation of the TCOE. In the coming years, highly effective organizations will continue to use their TCOEs for reaping the benefits of innovation and gaining a competitive advantage in the field of testing

services. Organizations will be able to integrate improvements in process and technology for improvements in the financial, customer satisfaction, and business related dimensions using the power of Testing Centers of Excellence.

> biography

Nishant Pandey
works with Capgemini and is based in Toronto, Canada. He manages Business Analysis and Testing engagements for clients in the Financial Services domain. Nishant is a certified PMP® and holds the Advanced Level Certificate in Test Management from ISTQB®. Nishant's articles on software testing and test management have been published in Agile Record, Testing Experience and Quality Matters magazines. Recently, Nishant has authored a chapter on Benchmarking in the book 'The IFPUG Guide to IT and Software Measurement'.



Transformation To A Test Center Of Excellence (TCoE): Decentralization To Centralization Approach

by Venkatesh Sriramulu, Ramakrishnan Subramanian, Prasad Ramanujam & Swaathiga KN

In today's dynamic economic environment, the complexity of various software and systems that businesses deploy is increasingly making it imperative for organizations to modernize their quality assurance and management practices. IT organizations are increasingly being asked to ensure an early ROI while delivering improved quality at lower risks for strategic IT related investments. Complex business requirements, highly diversified technology platforms, reduced time to market obligations and increased quality assurance demands have required testing organizations to bring about a change in the testing culture.

The need of the hour is to establish a proactive, robust and scalable testing platform in an organized and structured manner. The growing trend to meet all such needs is by establishing a Test Center of Excellence (TCoE) which renews the whole process of testing and empowers the organization to showcase continuous improvements in terms of quality, cost and time. Today, a growing number of enterprises already benefit from taking the TCoE route and they bear testimony to the fact that the future for TCoE holds immense potential. A recent Market-Pulse survey conducted among CIOs from various companies revealed that over 68% of respondents were either already leveraging or very likely to leverage TCoE as part of their testing delivery model.

The Transformation: Traditional Testing To TCoE

Global demand for high quality, low cost IT services from customers has created a significant opportunity for IT service providers who can successfully leverage the benefits of, and address the challenges in using, a robust and well structured testing platform across the organization. In the traditional approach of testing, the focus of the quality assurance team is to ensure that the system under test is in accordance with the business specifications. Organizations usually have dedicated testing teams, dedicated infrastructure and all the tools in place to achieve tremendous success on a short term vision. On a long term vision, however, such organizations often fail to realize that they exceed time and budget to promise such quality products. The reason is not that we lack in innovating solutions for quicker implementations and testing, but that we miss to adopt solutions that are readily available and proven to be successful. In order to achieve operational excellence and address the client's business needs more effectively, organizations are continuously rethinking and transforming their testing delivery model.

"A TCoE is a centralized testing model that brings together people, process and infrastructure into a shared services function for standardizing processes and optimizing resource utilization in order to generate exponential benefits across the organization."

Organizations which have succeeded in implementing TCoE opted for a long term plan. They didn't only find solution to a problem at hand but also decided on strategies like improving the returns on investment and optimizing the time to market, which are more important in business relations. Setting up a TCoE is NOT an easy job! It demands synchronized effort and a proper communication channel among its different teams, so that efforts can be focussed on scaling a pilot CoE with a small expenditure and minimal effort into a full-fledged TCoE by adding resources, capabilities and services once it starts giving the necessary returns to our business. In order to set up a successful TCoE, we need to think through the following routes:

- Standardize the test process within the test organization
- Frame a resource utilization matrix to monitor and bring up solutions for resource optimization
- Manage and measure the test effectiveness across the organization
- Bring up a shared model to share test assets, knowledge, best practices and other resources
- Set up single-point ownership for each specialization area of testing, such as performance testing, test automation, etc.
- Establish industry benchmarks and best practices. Document each process and all the procedures that are important to our business
- Conduct periodic audits to check for the compliance to standard practices

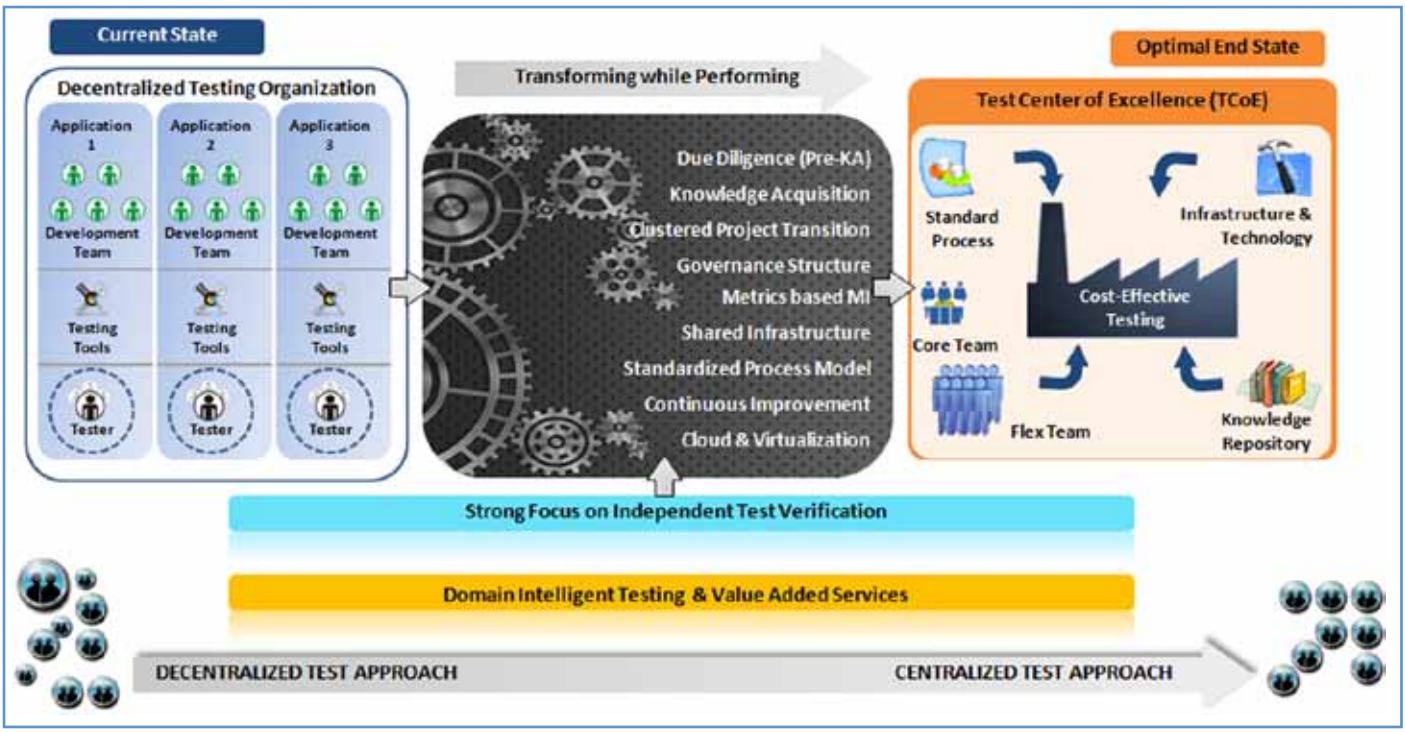


Fig. 1: Transformation of Testing To Test Center of Excellence (TCoE)

Building Blocks Of TCoE

People, process, infrastructure and technology are the key to success of any transformational change initiative, including TCoE implementation. The organizational objective should be in alignment with all four parameters to achieve the TCoE quicker and reap the complete benefits of establishing a Test Center of Excellence.

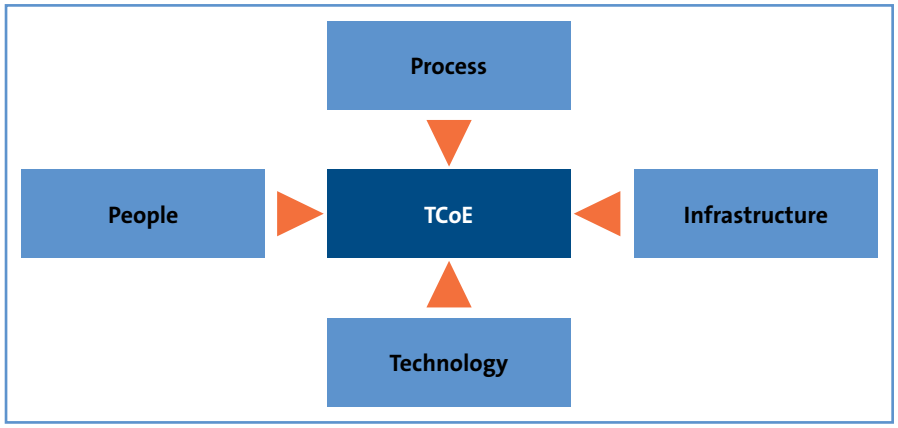


Fig. 2: Building Blocks of Test Center of Excellence (TCoE)

People

Most of the testing teams struggle to access a scalable and flexible resource pool to meet the peaks and troughs associated with the business requirements. Proper resource and release planning is highly important as it helps in formalizing processes and developing measurement systems to evaluate the project size from a human capital perspective. With frequent releases and resource alterations, knowledge retention is almost a non-achievable factor, and every team tends to spend their effort on the same kind of challenges as the lessons learned and solutions are not industrialized. Below we listed a set of drivers and processes that help in overcoming the difficulties associated with these problems.

- **Core/Flex Team**: This resourcing model is constructed to increase operational efficiency by retaining a knowledge base that can be utilized across projects. Core members who are experts in particular areas of testing and business domains drive the team to achieve the required output by eliminating the duplication of effort spent on finding the existing solutions by educating the Flex Team members who are deployed and released based on project specific staffing requirements.
- **BA-QA Model**: The QA team will be shadowed by a BA, who possesses both domain and testing knowledge, and together they derive a structured knowledge management plan in order to address the knowledge gaps. They will be engaged in various activities such as analyzing requirements, scenario creation, test case review, defects validation and provide support for challenging projects.
- **Capacity Planner**: It helps the test management team in determining the production capacity required in terms of people, resources, infrastructure and software by an organization to meet changing demands for its services. It is highly crucial for a capacity planner to make trade-offs between the current demands and future forecasts in a cost effective manner.
- **Knowledge Management Framework**: Availability of a generic knowledge management framework or reusable scenario repositories will facilitate faster access to knowledge and improved performance. It is a centralized repository used to enable testers to acquire, package and share knowledge on various projects, process and metrics to enable decision making, innovation and communication.

Process

A root cause of testing inefficiency is multiple processes being used by various testing groups in an organization. This inefficiency produces inconsistent results, prevents effective communication between different groups, and makes it difficult for senior management to understand where things are going well and

where they are not. Standardization of processes yields immediate benefits of predictability and is a good first step to take without introducing risk to our existing testing operations and need to maintain the flexibility to adapt the new approach and capabilities across different project teams. To aid such standardization, few key enablers are necessary which can be enumerated as below.

- **Global Delivery Platform**: It is an integrated end-to-end test management system that complements the testing delivery model to deliver sustainable business growth. It addresses many aspects that are critical to any project delivery and strives to establish common testing process templates, standards and benchmarks across the industry by promoting a cost effective and pay-per-use system.
- **Metrics Driven Approach**: It is a structured measurement program which allows identifying the areas for performance improvement, benchmark against industry/competitors, set targets, identifying trends for forecasting and planning, evaluating the effectiveness of changes, determining the impact of project management and creating perception about project performance across the organizations.
- **Innovation Ecosystem**: It provides a set of innovative value-added services for each phase of testing which work together as a single cohesive platform to facilitate testing activities and delivering excellence. It is executed by a centralized client-server model, and individual test teams have the ability to customize the value-additions as per their needs, thereby accelerating the testing cycle.

Infrastructure

Large enterprises often have many test environments, various testing tools, and separate environment support groups with different policies. This differentiation across groups leads to poor utilization of test assets, high support costs, and inefficiencies in tracking and managing test assets. Centralization of test assets and creation of a centralized testing infrastructure management group leads to an immediate improvement in asset and resource utilization, which would achieve significant cost savings.

- **Shared Services**: On-demand shared services delivery platform capable of provisioning modularized testing services including application testing, non-functional, test automation, tools and infrastructure rapidly thus reducing total cost of ownership (both capex and opex cost) with greater flexibility and agility.
- **Managed Test Center**: It is designed to support the peaks and troughs of quality control and testing requirements, leaving the customer to focus on his core competence. The full ownership of the quality control activities are shifted to the service providers by enabling greater risk sharing and cost and resource optimization.

Technology

Contrary to what many believe, testing usually ends up being a time consuming activity that affects time to market, unless the testing process is improved. The most obvious reduction in cost with improved quality comes in the form of adopting modern technologies such as cloud and virtualization, backed by solid technology and alternative engagement models that are best for automating the testing process, which would reduce the human resource costs required for the quality output. Listed below are certain technologies that drive improvements in testing:

- **Automation CoE**: Test automation is a sure method to reduce time to market by compressing the testing life cycle. It is important to quickly pick up all the possible candidates for automation and to reap the benefits of automation, not only during the maintenance phase but also at each stage of the testing process.
- **Cloud & Virtualization**: By deploying technology as a service (test management tools, test environments and other tools) and virtualizing the application in the cloud, the users access only the resources they need for a particular task, which in

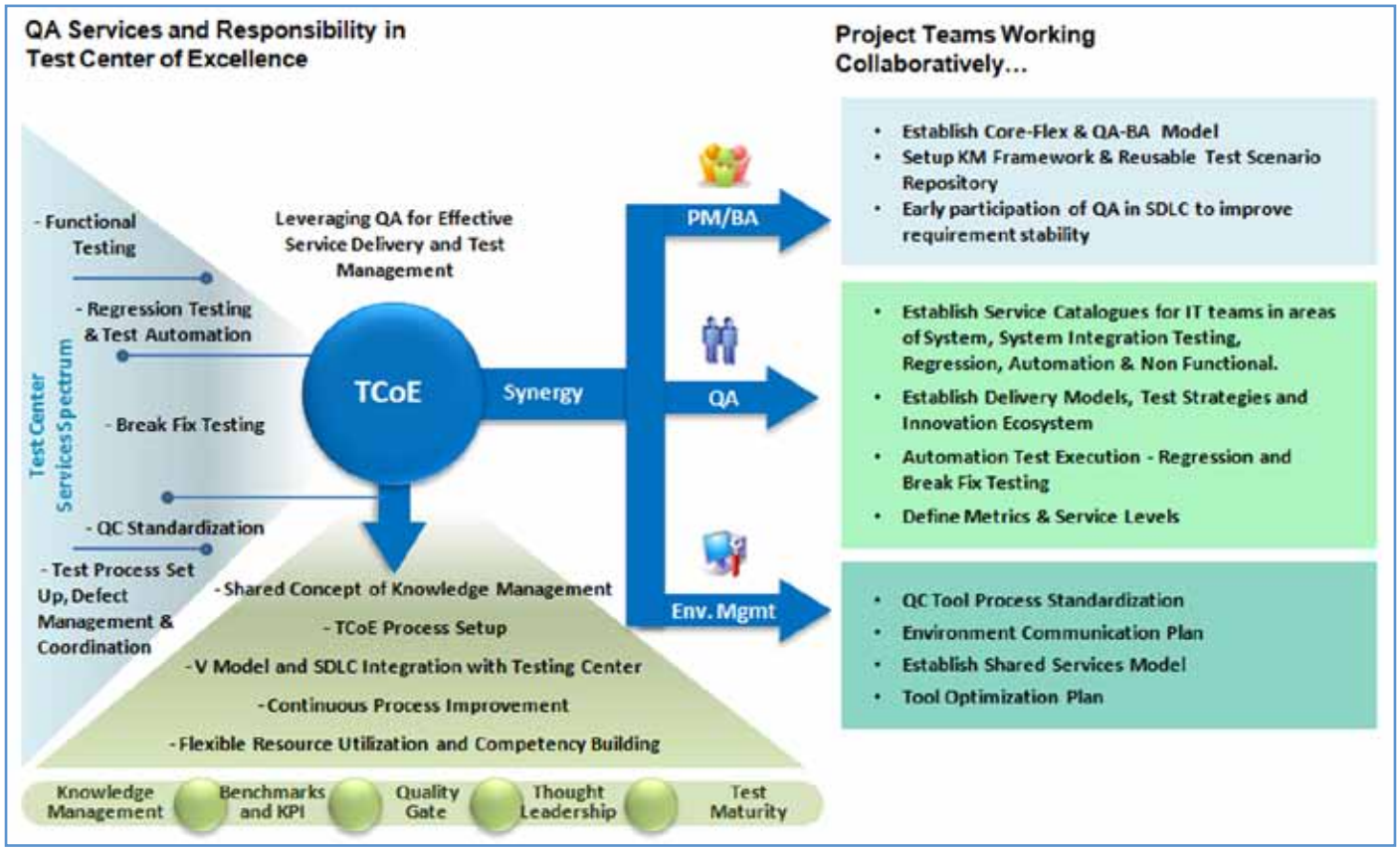


Fig. 3: Leveraging TCoE for Centralized Testing Approach

turn prevents having to pay for idle computing resources. It will greatly reduce the licensing and associated costs of software tools to a greater extent.

- **Risk Based Approach**: It allows the selection and prioritization of the test suite based on features, functions and probability of risk prone or failure areas, in order to limit the testing of mathematically large combinations of test cases to a small subset with 100% test coverage.

Leaveraging TCoE For Centralized Testing Approach

(See Fig. 3)

Challenges Faced In Implementing TCoE

- Most of the large organizations are geographically dispersed and this often leads to a lack in communication, ineffective resource utilization and operational inefficiencies. At initial stages of the transformation, it is hard enough to prioritize projects across different units, to identify if there are any duplicated efforts and investments, to mitigate risk and find opportunities for collaboration amongst locations across the enterprise.
- An important challenging factor is to standardize processes, methodologies, project management practices and tools across the organization. This requires a better synergy from different project units to realize and deliver what went wrong and what went right before establishing the organizational benchmarks.
- A TCoE is led from the top down and implemented from the bottom up. Leadership from the top is necessary to define, prioritize and construct various competency centers leading to a TCoE. However, its benefits cannot be derived unless everyone is empowered and given the knowledge and skills necessary to take action and improve their competencies.
- To adopt the test management and test automation tools which should satisfy the needs of various projects dispersed across the wide organization requires thorough assessment and informed decisions by skilled test engineers.

Benchmarking TCoE

To ensure the TCoE's continued success, establish the base metrics by which we can measure the effectiveness of the implementation. These metrics will serve as key performance indicators that can be used for monitoring, reporting, and overseeing the program and to measure its impact on the organization. We recommend defining metrics in two categories:

- **Program Metrics**: It measures the success and progress of the overall program throughout the organization and the quality of the services provided, and it also measures and compares the actual results of the costs and benefits for the program rollout vs. the original business case. Metrics in this category include the number of concurrent projects, costs of the program, solution adoption, user adoption, demand pipeline, and implementation time.
- **Operational Metrics**: It measures the availability, stability, and responsiveness of the deployed systems; the quality of the system services; and the capabilities deployed to business units and the adherence to established service level agreements (SLAs). Metrics in this category include uptime, access and content retrieval performance, capacity utilization, transaction volumes and performance, and customer satisfaction.

Metrics will provide the historical data which need to develop cost/benefit analyses for future application design and implementations. Performance metrics can be used to evaluate alternatives for business process optimization, while customer satisfaction metrics can be used as input for continued improvement.

Benefits To Customer And Organization

- Faster Time to Market and Reduced Cost: TCoE brings together a best-in-class of career testers to achieve best-in-industry testing capability enabling the customers to hit the market soon.
- Flexible Resource Pooling: It enables enterprises to scale up/scale down the testing team in synchronization with the various Business and IT initiatives.
- Improved Quality: It delivers a mature process oriented approach towards testing, ensuring standard test methodology and leveraging best practices to improve test quality and effectiveness over a period of time.
- Reusability of Test Assets: Centralizing testing best practice processes and tools guarantees for continuous improvements to the customers in terms of cost optimization.
- Process Standardization: Standardization of process enables an enterprise to become a more transparent and flexible organization, and can be the basis for realizing operational excellence by achieving high predictability.
- Productivity Improvements: Committed YoY productivity improvements and cost savings due to the well-defined testing process and best practices used across the industry.
- Estimation Prediction: The metrics driven approach to testing enables accurate and factual prediction of resource and cost estimation with ease.
- Risk Sharing: With a centralized testing approach, the risk of project delivery can be shared across all the stakeholders.

Conclusion

One of the key questions that every CIO should ask is: "Why should we move from the existing state to a TCoE?". The answer is: "To organize the endeavour by creating a general set of resources, procedures, processes and tools to meet the extensive set of business requirements while taking benefit of existing organization competence, standards and infrastructure". It also helps to reduce and optimize technology expenditures, raise organizational effectiveness due to shared resources; acceptance of regulations in a faster way, and improve on overall business flexibility. Thus it creates a vast repository of shared knowledge by virtue of which the learning curve for the employees can be increased. TCoE helps organizations to standardize their quality best practices that assist in maintaining consistency, efficiency, and rapid implementation of superior processes. The Test Center of Excellence caters to the testing needs of enterprises around the globe while lowering QA and maintenance costs and enabling tighter alignment of IT with business objectives.

Case Study

Establishing A Testing Center Of Excellence For A Fortune 500 Financial Services Organization

The client is a Fortune 500 financial services organization offering a broad range of financial products and services including life insurance, annuities, mutual funds, disability income insurance, bank products and more. The client's IT group felt the need to consolidate their QA practice and reduce operating costs while increasing efficiency and consistency within its software quality organization. The current QA team lacked the required domain

and system knowledge, and there was no formal knowledge management process in place.

The client invited a QA organization to undertake an assessment of their landscape and propose a road map for setting up a QA background which would address the needs of the growing business while aiding the following three key change imperatives: 1) Faster Project Execution, 2) Resourcing Model to accommodate future spikes, and 3) Maximizing the value from testing.

The Solution

The QA organization conducted an independent assessment of the client's QA landscape and proposed the establishment of a consolidated QA set-up, a Test Center of Excellence. They proposed a clear organization structure for the TCoE and developed the roadmap and partnered with the client in standardizing and centralizing QA activities within the TCoE.

Business Benefits

- The established staffing model using core and flex teams ensured dynamic resource management
- Increased visibility to testing activities through comprehensive metrics collection and reporting
- Significant productivity improvements by process standardization and sharing of best practices
- Reduced IT cost through elimination of multiple/redundant products and infrastructure
- Effective training process (50% effort reduction in Knowledge Transfer) along with efficient job rotation to manage the ramp-ups in short timeframes
- 44% effort reduction achieved through automation
- 30% reduction in overall IT budget

Acknowledgement

We wish to express our thanks to Mr. Venkatesh Ramasamy, Project Lead, Cognizant Technology Solutions and Miss. Alisha Bakthawar, Senior Test Analyst, Cognizant Technology Solutions for their dedication and worthy researches in TCoE implementation study and analysis, that has helped in shaping this article and bringing it to fruition.

> biography

Venkatesh Sriramulu
currently works as a Project Manager at Cognizant Technology Solutions, Chennai. He has 8+ years of experience in the IT industry focusing on software testing and project management all through his career. He has excellent experience in managing end-to-end enterprise IT project life cycles and set out roadmaps for streamlining and introducing testing tools in complex business engagements, which is a driving force in improving software test processes. He is also specialized in Agile and SOA testing and has come up with many innovative ideas for the improvement of testing process.

Ramakrishnan Subramanian
currently works as a Senior Project Manager at Cognizant Technology Solutions, Chennai. He has expertise in building and leading world-class quality assurance functions, and in championing enterprise-wide process and capability improvements. He is vastly experienced in working with large insurers and financial services organizations, in particular for setting up independent testing for their enterprise level testing needs. He is passionate about instilling a culture of quality in technology organizations and has been a proven business partner in diverse insurance and financial industries.

Prasad Ramanujam
currently works as a Senior Project Manager at Cognizant Technology Solutions, Chennai. With his 11 years of experience in the IT industry, he is a recognized leader in agile methodology and a frequent speaker at industry conferences on the subject of successfully implementing agile and SOA testing techniques in organizations. He has supported many multi-national clients in various industries in becoming high-performing businesses by establishing a structured and strategic approach to Quality Assurance.

Swaathiga KN
is an ISTQB® certified testing professional, and currently works as a Senior Test Analyst at Cognizant Technology Solutions, Chennai. She is specialized in end-to-end testing and developed many software products for doing end-to-end test management activities to optimize the testing costs and improve the application quality. She is also involved with many organizational process improvement activities to bring out the best possible solutions for Quality Assurance.

Prof van Testing recommends

Díaz Hilterscheid

twitter

Follow me @vanTesting

IREB –
Certified Professional for Requirements Engineering – Foundation Level

Description

The training is aimed at personnel mainly involved in the tasks of software requirements engineering. The tutorial is designed to transfer the knowledge needed to pass the examination to become an IREB CPRE-FL.

After earning a CPRE-FL certificate, a certificate holder can assess whether a given situation calls for requirements engineering. He understands the fundamental characteristics of the discipline and the interplay of methodological approaches, e.g. interview techniques, description tools or forms of documentation.

More information regarding the required knowledge can be found in the IREB Syllabus, which can be downloaded from the IREB web site: http://www.certified-re.com

IREB International Requirements Engineering Board

Dates*	3 days
06.–08.06.12	Berlin (de)
19.–21.06.12	Oslo/Norway (en)
09.–11.07.12	Mödling/Austria (de)
25.–27.09.12	Mödling/Austria (de)
16.–18.10.12	Berlin (de)
16.–18.10.12	Stockholm/Sweden
27.–29.11.12	Berlin (de)
27.–29.11.12	Oslo/Norway (en)
27.–29.11.12	Helsinki/Finland (en)
18.–20.12.12	Berlin (de)

*subject to modifications

Website:
http://training.diazhilterscheid.com

Dependencies walkthroughs – footprints to effective testing

by Jacky Kayan Cheung

Challenge in service automation projects

Traditional mechanical processes are commonly automated by computers and electronic control. Such service automation products and systems increase with their complexity, and they require enormous effort for testing. These systems require both hardware and software testing. Government and public organizations, such as railway companies, hospitals and postal services, are the typical customers of automation systems. They request high standards in documentation, development and manufacturing control, in order to ensure system reliability and consequently a public image characterized by trust. Automation procedures and requirements depend closely on regional needs, and product customization is in high demand. In this situation, it is very difficult to get testing staff with the specific domain know-how, in order to work proficiently with customers and stakeholders. There are substantial differences in the development and manufacturing cycles among computing, electronic and mechanical engineering disciplines. This fact introduces extra effort for the test team to enable them to interact with inter-department needs within a project. Altogether, these aspects can present an enormous challenge for every team involved in the project. To overcome this, we perform walkthroughs of some domain specific factors together with the test personnel, whereby some of these elements are applicable for general purpose as well.

Setting verification at origin

Planning for a multi-disciplinary project is always difficult. Activities need to be coordinated well in order to guide the tasks for overall project efficiency.

In view of the high complexity and closely-linked cross-disciplinary dependencies, it is vital to have test analysts join the project in the early stages. Some organizations think that testing takes place in the final stages of the project, and that testing staff are only required for the late project stage. In reality, however, the requirement-to-specification high level design process can be verified at an early stage so that early availability of specifications can be promoted. For projects with a substantial amount of newly developed modules, it is crucial to have a test analyst with the related domain knowledge. This way, specification verification can be secured by identifying major uncovered requirements as early as possible during the design phase, rather than facing exhaustive rework towards the operational phase. Meanwhile, writing of use cases and functional description can be enforced, so that testers that join at a later stage can quickly acquire the necessary domain know-how, which minimizes the training effort.

Establishing cross-team collaboration

Test case writing forms an essential part in the test preparation phase. This task is especially difficult because of the multi-disciplinary nature of the service automation domain. Employing testers with multi-discipline experience of computing / electronic / mechanical engineering would help substantially. Concurrently, collaboration between testers and developers plays a crucial role. This can be accomplished by workshops, e.g. to discuss functional specifications. Such processes can contribute to synchronizing project terminology and create documentation across multi-disciplinary departments. Project members, who usually come from diverse disciplines or domain backgrounds may have different interpretations for the same vocabulary. It can save a lot of trouble to harmonize the common project language.

Besides all this, it is well worthwhile for testers that joined the project early to participate in various development support activities during the early project stage. They can thus become familiar with the project through this hands-on experience, which can subsequently help in setting up the test facilities. Meanwhile, developers can benefit from an early verification of their proposed concept / model. This could be especially helpful for modules that are being newly developed. Moreover, it can also help the mutual understanding between testers and developers, and will provide an effective work environment for the testing phase.

Planning workflow and resources

Planning of test execution is usually a difficult task. This task requires additional effort for projects with completely new hardware design. In such cases, intensive hardware testing is required. As the hardware corrections require time, it is sensible to coordinate the critical paths between test execution and the hardware develop-

ment / manufacturing cycle. Project workflow can be streamlined according to the inter-module dependencies; e.g., some high level software modules would depend on the readiness of the associated firmware, and similarly some firmware may depend on its associated hardware. Testing of modules would therefore depend on the corresponding module readiness.

During the integration and test phases, a lot of resources are used. Some test facilities come with internal dependencies. Prototypes, for example, may need version upgrades in different iterations. Quantities of spares and corrections have to be planned for due to the low reliability in developing prototypes. Some other facilities come with external dependencies, such as interfaces of external servers. External organizations also often share their test server environment within other entities, and test configurations would therefore need to be adapted for various stakeholders. This situation requires the schedule to be coordinated with external organizations in order to ensure that the test resources are available, and that the required test configurations can be provided accordingly.

Focusing on high risk items

Both software and hardware development usually involve a number of iterations within a project. This leads to software-hardware dependencies, e.g., a high level requirement violation may be indicated by an application software test case failure, which eventually requires a hardware change. Such hardware changes may imply associated software changes for the next iteration. The focus should first be on high risk modules so that critical modifications can be implemented during the earlier iterations. Prompt testing of such critical modules would frequently help in reducing the risk of overruns in iterations. The testing of less critical modules can in the meantime be given a lower priority.

Complete coverage vs. complete overview

Having comprehensive test case coverage of a project in the service automation domain is not an easy mission. Multi-disciplinary and specialized domains make it difficult to find a single tester with all the know-how required. Instead, extensive know-how of the entire test team can be achieved by combining testers from different technical disciplines and domain-specific know-how.

Quite often test projects would come with some peak and off-peak periods within the project life. Making use of the off-peak periods, testers of different modules can conduct knowledge sharing, so that each member can gain a complete overview of the systems and acquire a broad knowledge base. It is generally useful for test case authors to have an overview of the system, so that test case design of the various modules can be integrated, and untested gaps can be minimized.

Complexity of automation systems is generally high, so 100% test coverage is not attainable in terms of effort. During test case design, it makes sense to outline the overall picture of the possible scenarios, and to put the important scenarios into test cases. This gives an overview of untested scenarios as well. When unexpected software bugs appear, such a complete overview is helpful to identify the missing test scenarios, and enable new associated test cases to be quickly developed.

> biography

Jacky Kayan Cheung
is a senior system test engineer at PASS Technologies AG in Zürich. Recently, he has started to test medical instruments. Prior to this, he served ACS (A Xerox Company), Thales and Hong Kong MTRC for ticketing systems and control systems in the transportation business. He has worked on international projects in Hong Kong, Taiwan, Melbourne, Dubai and Switzerland since 1995. With both an electronic engineering and a computing background and strong interests in embedded control systems, Jacky has been involved in the complete project life cycle and recommended system improvements. Jacky developed his profession in user acceptance management and has specialized in system test since 2008.

Setting up a Testing Competence Center of Excellence: Making the Process Personal

by Thomas Veltman

Years ago when I was first started working in testing, I was involved in a project at a large pension fund manager. I worked in a particular project where we needed to create a lot of test clients to be able to execute our test cases. One of the problems we had to solve to achieve this was that the particular identification number for clients (called "KlantID") had a lot of constraints – you could not just use some random string, it had to fit a certain format.

A lot of testers have been in a similar situation, and most of them will create an Excel sheet or some other type of small application to generate this kind of KlantIDs. It is just the logical thing to do, and so that's what I did as well. It took me about a day to build and test this (unfortunately it wasn't right the first time), but when it was finished, it was a very valuable tool which the project kept using for some time.

Then, after a while the pension fund manager decided to professionalize their testing effort, and one of the steps they took to achieve this was that to centralize their testing and organize all testing activities in a testing Competence Center of Excellence (TCoE). One of the reasons for this step was that all testers could share knowledge more easily and make efficient use of the available testing tools.

So when the center was started up, we found that there were three different kinds of sheets like mine to generate Klant-IDs. So this was a quick win for the CoE – we could take the good points of the different sheets and merge them into one new one. This is exacly what we did, and we started using this tool across all projects.

A year later, a tester who was new to the organization came to my desk with a question. Of course, a TCoE is a very dynamic organization, so people join and leave every month, which means there is also a lot of knowledge transfer going on at all times.

The new tester, who had created an Excel sheet to generate "Klant-IDs" which he had been using for the past month, now wanted my opinion on it. We had the feeling that some of the work had been done twice, so in the next team meeting we asked who was aware of the fact that there was a tool especially created for this. It turned out that most of the newly arrived testers (and even some of the exsiting testers) were not aware of this tool; it even turned out that another new Excel sheet had been created!

What we can learn from this story is that even when testing is centralized, it is still not easy to organize knowledge sharing and communication in such a way that resources are used most effectively. The main reason is that setting up a Testing Center of Excellence is not an easy task – it requires constant attention and it is a lot of work. This is supported by statistics: According to the 2011 World Quality Report, only 4% of companies worldwide have a Testing Competence Center of Excellence operational, and an additional 6% are in the processes of setting one up. This is a remarkably small number as the concept has been around for around ten years and many companies have the desire to set up a Testing Competence Center (around 46% of clients according to the WQR).

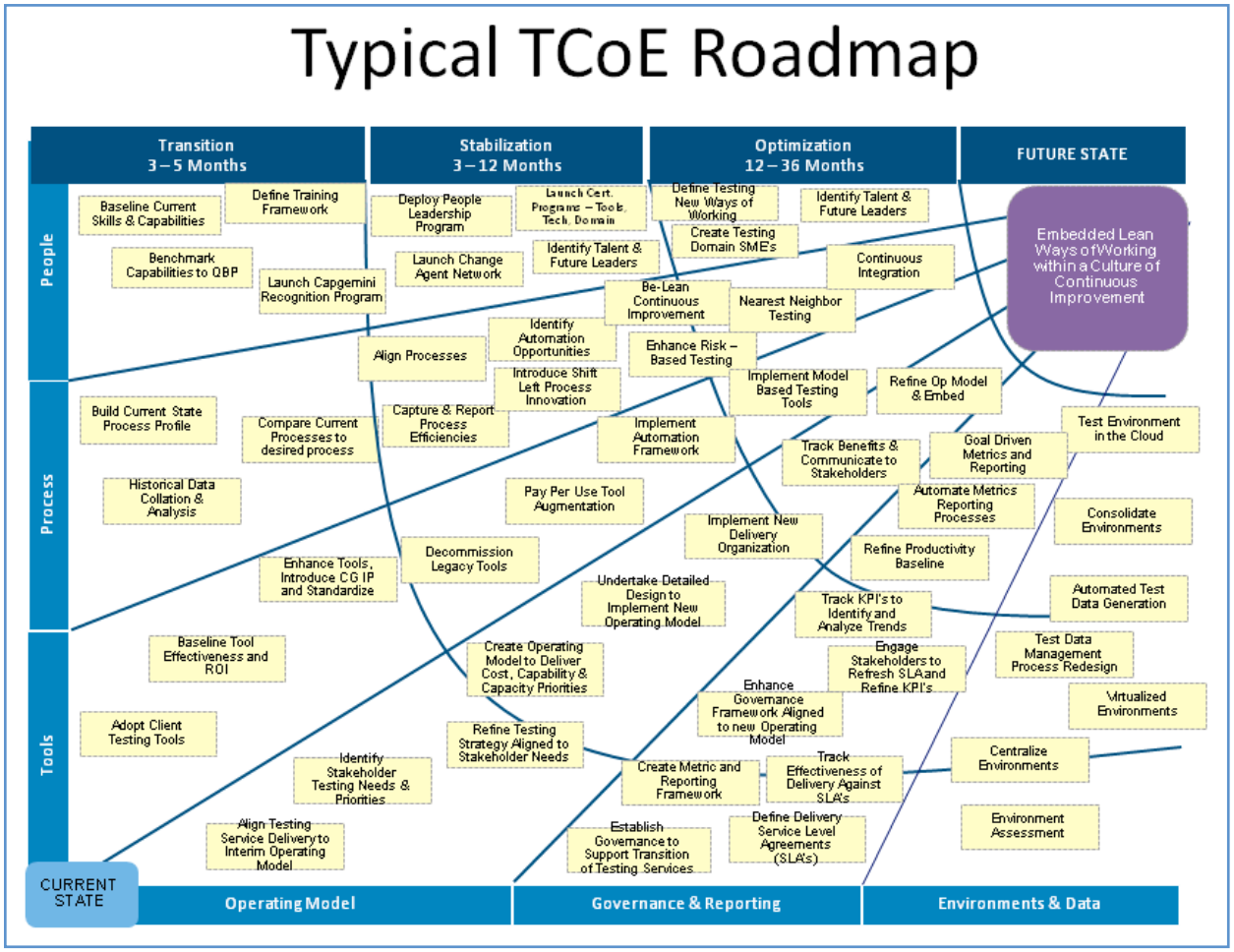
Setting up a Testing Center of Excellence

So we know now that it is not an easy task to set up a competence center of excellence. How can we make this task easier and how should we go about it?

The answer to this question is to first and foremost get the answer to another question. This question is: Why do we want this competence center of excellence? There may be many reasons:

- To be able to use tools centrally and save costs that way
- To be able to use test capacity effectively and save money that way
- To be able to share testing and domain knowledge effectively between testers
- To professionalize testing and create a better environment for testers to work in
- To be able to outsource (parts of) the testing process to another location.

Once the goals are clear, a start can be made with setting up the center. Generally, it is best to designate the first period of setting up the TCoE as a transistion period during which the following six characteristics are put in place for the TCoE:



- **People**: Getting the right people with the right knowledge and skills to work in the right place in the TCoE, and making sure that it stays that way.
- **Process**: Having the most efficient testing method
- **Tools**: Making sure that tools are available to support the testing method optimally
- **Operating model**: Making sure that the TCoE is organized well and fits well within the client organization.
- **Governance and reporting**: The performance of the TCoE must be measured and reported about, so the stakeholders for the TCoE must have sufficient tools to be able to manage the TCoE efficiently.
- **Environments and data**: Ensuring that the test environments are available to testers when needed and that environments and data are managed in an efficient way.

The kind of solution that is needed for each of the characteristics depends on the reason for setting up the TCoE. For instance, if we design a KPI report, it is important that we include KPIs that measure the goals for the TCoE. If one of the goals is to professionalize our testing, we should include a report that reports what percentage of testers in the CoE have a certification of some kind. If one of the reasons for starting the center is offshoring, then we should take this into account when designing the operational model as some of the activity will be done in another country. Of course an offshoring decision also impacts other characteristics, e.g., test environments and how we can make these available on the other side of the world.

Setting up the center takes time. Generally, a transition period of nine to twelve months is needed, but this may of course vary for each organization. It is very important to take this transition period seriously though. Many TCoE implementations fail because this time is not taken to really build the center.

Spread the word!

Besides the characteristics of the testing center, it is important that a lot of noise is created around the TCoE to let people know that the TCoE is there and that projects can use the TCoE for all their testing and quality related questions. So spread the word and make clear that the objectives for the TCoE are being reached. Focus on quick wins.

One of the quick wins that you will see quickly in a TCoE is the fact that tester waiting time will be reduced. In a typical project, the delivery of software and test environments is usually late. The testers, who are generally ready to start when the software should have been delivered, have nothing to do if delivery of the software is postponed. So they wait. This of course costs a lot of money for the project. In a TCoE many projects are running in parallel, so if a tester is forced to wait for software in one project, there is always another project where his or her skills can be put to good use.

Do not stop building: optimize

From my example with the data generating Excel sheet, we learned that in a TCoE you can never take for granted that everything that is created will be used in the right way. There are always new tools and ways of working to incorporate, and you have to point out to

testers which tools and best practises are already available. This has to be periodically evaluated, adapted and communicated to all the testers. In many cases, a TCoE starts with a limited number of projects and expands gradually to incorporate more. So the optimization of the TCoE never ends, but fortunately with a TCoE it is much easier to keep up to date with a lot of things than it is with testing activities scattered throughout the organization.

Make it a fun place to work

In general, organizations can be resistant to change and this can also be the case when you are setting up a TCoE . It is important to have the testers involved in setting up the TCoE. The TCoE will make access to tools easier, there will be less waiting time for testers, and a TCoE will generally create more visibility and status for testers. All these are things that testers like. So if they can see that this change is for once a change in their favor, they will have a positive influence on the process of setting the TCoE up. In my years as a tester, I have worked in two TCoEs, and they were great places to work as a tester – you can learn many things from your colleagues and you get see a lot of different projects.

A personal process

As said before, setting up a TCoE is not an easy task. In fact, it takes a lot of skill from all the people involved, from the testers, from the manager of the CoE, from the people working with the TCoE. There is no quick solution for this. However, if I could give just one advice, it would be this: It is easy to get lost in all the processes that surround a TCoE. Do not focus all your energy on creating the best processes, templates and tools, but invest in the people that form the CoE. If you can get the testers involved early, you have many ambassadors that will be ready to pick up tasks in every way possible. Make sure that the processes are personal!

> biography

Thomas Veltman *is a test consultant working for Sogeti in the Netherlands. He has been working in Test Competence Centers of Excellence as long as he's been working in testing and has fulfilled many different roles in these – he was involved as a tester, a test manager and as a consultant in charge of setting up and optimizing a Testing CoE.*
He has been working for clients in many different industries (health, pensions, government, energy & utilities).
Besides working at clients, Thomas also works as a business developer in testing. His specialities are mobility & apps, and usability. He was a speaker at last year's Swiss Testing Day and at several Dutch conferences such as the Testnet events and the TMAP Day. Last year he was the team leader of the team that won the Cap Gemini/Sogeti innovation award in testing.

Díaz Hilterscheid

Online Training & Exam Preparation

ISTQB® Certified Tester Foundation Level (English & German)
ISTQB® Certified Tester Advanced Level – Test Manager (English)
ISTQB® Certified Tester Advanced Level – Test Analyst (English)
ISTQB® Certified Tester Advanced Level – Technical Test Analyst (English)
ISEB Intermediate Certificate in Software Testing (English)

Sample Questions & Answers, Exam Simulation, Quiz & Exercises, Videos

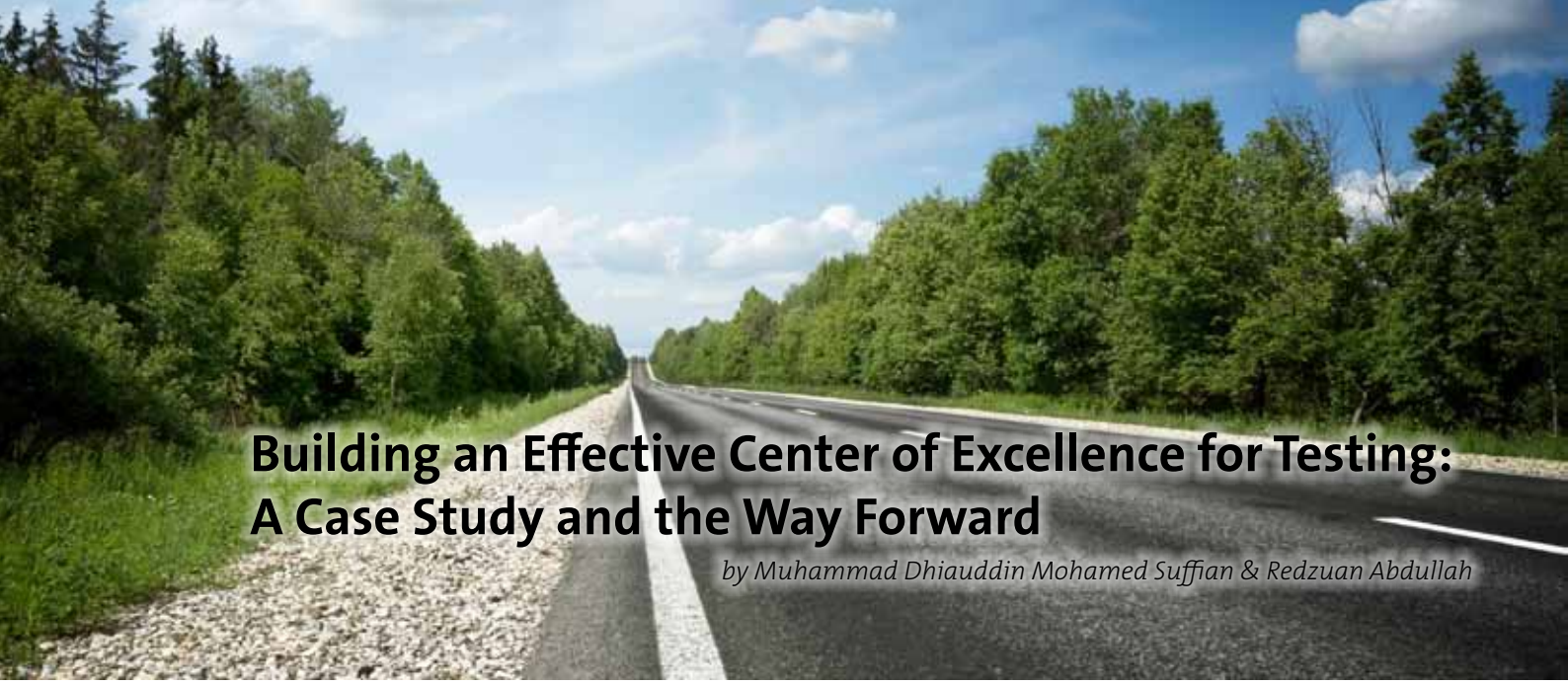
iseb Qualifications from BCS

ISTQB Accredited Training Provider

ISTQB 10 YEAR ANNIVERSARY International Software Testing Qualifications Board

Díaz Hilterscheid

te testing experience



Building an Effective Center of Excellence for Testing: A Case Study and the Way Forward

by Muhammad Dhiauddin Mohamed Suffian & Redzuan Abdullah

In an organization that does not compromise on the quality of the software and hardware produced, it is essential and crucial to establish a dedicated team that focuses on ensuring both products are sufficiently tested before releasing them to either the technology recipient or to actual end-users. This article presents the experiences made by the product quality and reliability department of an applied research and development (R&D) organization in setting up an effective testing organization by utilizing established models and techniques. During its first three years, the department played its role as a dedicated department that only executes system testing for software products developed by various labs in the organization. Later ittook over the additional role of conducting hardware testing. The department was established inthree major steps:

1. Adopt established model
2. Baseline current status
3. Develop improvement plan

Steps For Building A Center Of Excellence For Testing

1. Adopt Established Model

"Adopt established model" here means applying a common and proven approach in setting up the Testing Center of Excellence (Test CoE). The common model adopted can be represented in several elements: test environment, test process, continuous improvement, tactical and strategic dashboard, as well as stakeholder satisfaction. This is shown in Figure 1.

A brief description of each element is as follows:

- Test environment – the environment that is required for the successful operation of testing activities.
- Test process – a standardized process for conducting end-to-end test activities.
- Continuous improvement – embarking on various improvement initiatives towards strengthening test activities.
- Tactical and strategic dashboard – one-stop center for communicating test progress status and health status of software under test.

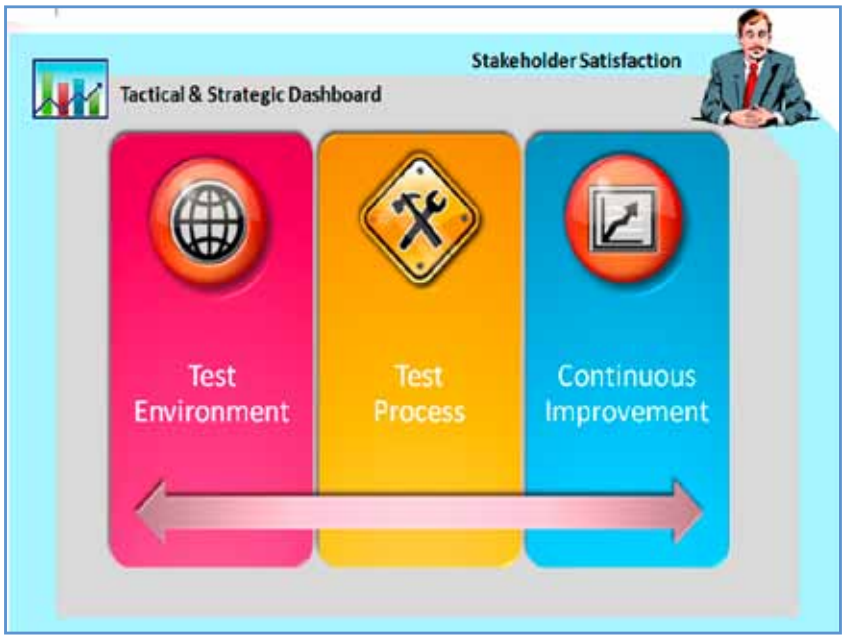


Figure 1: Adopting established model for Test Center of Excellence

- Stakeholder satisfaction – satisfying the management team that have the authority to release the software and the users that will use the software.

2. Baseline Current Status

In this phase, current capabilities and competencies are assessed to identify any gaps and the room for improvements. This is the point where the planning for improvement initiatives is put in place to fill these gaps. Two main areas are put into focus: test environment and test process.

a. Current Status: Test Environment

Assessing the current status of the test environmentmeans assessing the total or overall picture of the environment that is required to successfully implement test activities in the organization. Several aspects are taken into consideration:

- Test tools – assessing the tools available in-house, not just tools specific to testing but also the whole spectrum of the software development life cycle (SDLC) to support testing activities, ranging from requirements until the release and maintenance of software.
- Quality control – assessing current process and practices for quality control in developing good software.
- Test measurement – assessing current approaches and metrics to measure the test status such as test execution progress, defect status and overall health of the software under test.
- Use of test processes – assessing current implementation of test processes starting from test planning until test closure.
- Management support – assessing how far the top management of the organization value and acknowledge the importance of testing.
- User satisfaction – assessing the current level of satisfaction from users in using existing products and their expectations for future products.

Figure 2: Baseline current status of test environment

- Test environment planning – assessing the readiness and rigorousness of planning and establishing the right test environment for software under test.
- Test training – assessing how far training courses attended by test engineers have benefited the team and determine future training measures to equip them with the right competencies.

All of the above has been simply represented in Figure 2.

As a result, a chart is produced to clearly depict the current status or capabilities against areas for improvement. This is shown in Figure 3.

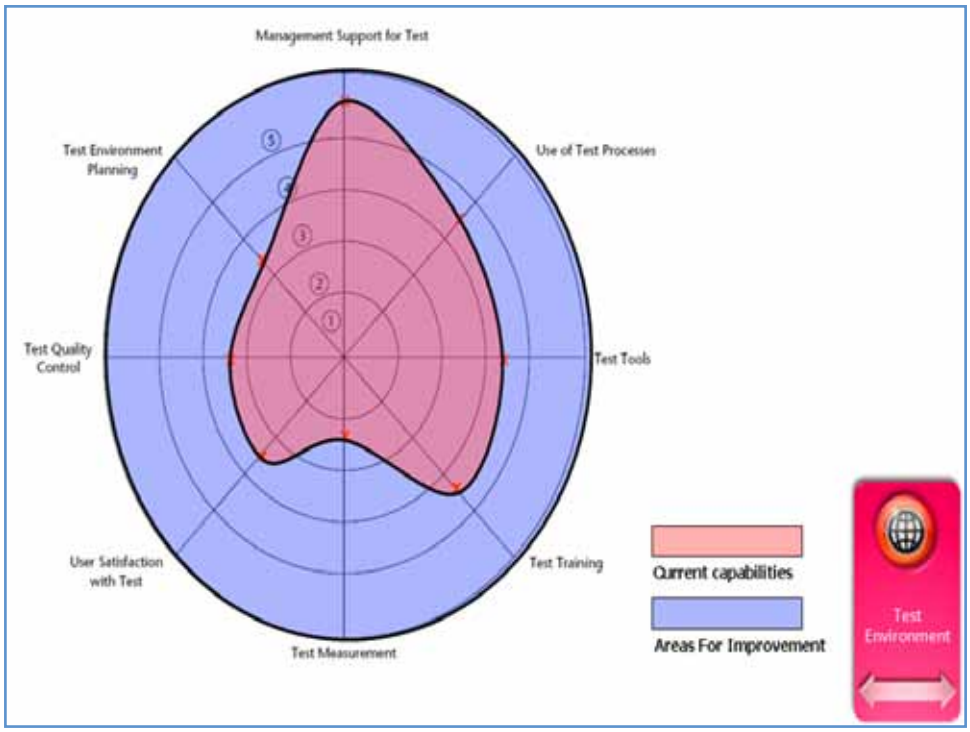


Figure 3: Current capabilities and areas for improvement for test environment

b. Current Status: Test Process

As for the test process, the emphasis is on assessing current practices and the implementation of the test process in governing the entire testing activities. This covers the following areas:

- Project preparation – assessing the involvement of the testing team during the initial phase of the project, and assessing in how far the team is ready with the resources and tools to support the execution of the project.
- Test planning – assessing the current strength and weaknesses of planning the test.
- Test execution – assessing the current strengths, drawbacks and limitations in executing the test.
- Acceptance testing – assessing the current process for performing acceptance testing with end-users or technology recipients.
- Analyzing results – assessing how good current test results are being collected, analyzed and measured.
- Reporting – assessing current ways and mechanisms of reporting the test results to various audiences, especially to project stakeholders and top management.
- Test types – assessing techniques and types of test used for testing the software under test that are of different nature and behavior.

Can agile be certified?

Find out what Aitor, Erik or Nitin think about the certification at **www.agile-tester.org**

Training Concept

All Days: Daily Scrum and Soft Skills Assessment

Day 1: History and Terminology: Agile Manifesto, Principles and Methods

Day 2: Planning and Requirements

Day 3: Testing and Retrospectives

Day 4: Test Driven Development, Test Automation and Non-Functional

Day 5: Practical Assessment and Written Exam

27 training providers worldwide!

Certified Agile Tester

We are well aware that agile team members shy away from standardized trainings and exams as they seem to be opposing the agile philosophy. However, agile projects are no free agents; they need structure and discipline as well as a common language and methods. Since the individuals in a team are the key element of agile projects, they heavily rely on a consensus on their daily work methods to be successful.

All the above was considered during the long and careful process of developing a certification framework that is agile and not static. The exam to certify the tester also had to capture the essential skills for agile cooperation. Hence a whole new approach was developed together with the experienced input of a number of renowned industry partners.

Supported by

Barclays
DORMA
Hewlett Packard
IBM
IVV
Logic Studio
Microfocus
Microsoft
Mobile.de
Nokia
NTS
Océ
SAP
Sogeti
SWIFT
T-Systems Multimedia Solutions
XING
Zurich

- Post-mortem – assessing the current process of doing post-mortem and in how far the testing team is involved in the process.

These aspects arerepresented in Figure 4.

Figure 4: Baseline current status of test process

Thus, the results of current status for the test process are presentedin the chart shown in Figure 5.

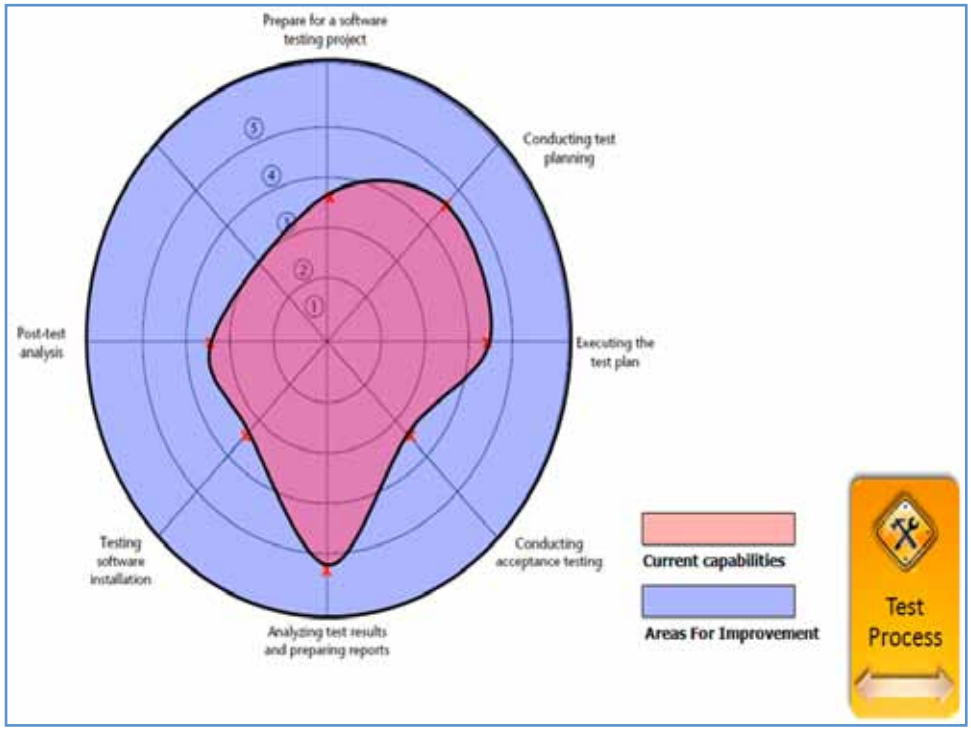


Figure 5: Current capabilities against areas for improvement for test process

3. Develop Improvement Plan

After the current status of the environment for testing as well as the test process have been determined, several improvement plans are established, developed and executed. These improvement plans can be divided into the following categories:

- Test involvement – Strengthening and improving the involvement of the testing team in all phases of software development: requirement, design, development, testing itself and system deployment. The testing team needs to be involvedin the kick-off meeting and requirement review during the requirement phase; they need to be involved in the design review and produce the test plan during the design phase; they need to generate test cases and/or test scripts as well as participate in code reviews (if possible) during development phase; they need to perform test execution, log defects and produce a test summary report during the testing phase; and they need to be involvedin the user acceptance test (UAT), beta test as well as pilot test during the deployment phase. This is presented in Figure 6.

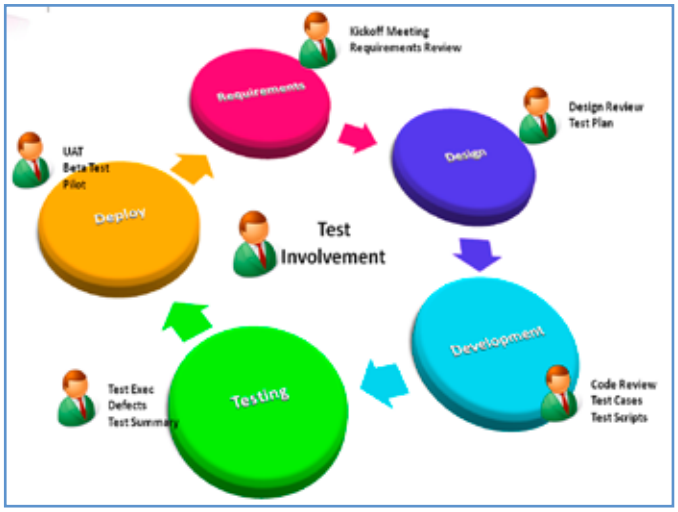


Figure 6: Improvement plan – Test team involvement in the SDLC

- Test process – Strengthening and improving the test process by ensuring the test process is standardized, unified and integrated into the whole application life cycle management (ALM) processes. This means besides SDLC process, it is also supported by effective project and quality management, configuration and change management as well as continuous process improvement, such as being part of Capability Maturity Model Integration (CMMI), Six Sigma and Test Process Improvement (TPI) initiatives. This is presented in Figure 7.
- Test strategies – Strengthening and improving test strategies or test types to cater for various types and nature of software, hardware and system to be tested. This is depicted in Figure 8.
- Test core team – Strengthening and improving the structure of the core team for testing. This is done by establishing a structure that allows for having a group of test specialists and sharing of test knowledge. This is depicted in Figures 9 and 10.
- Project involvement – Strengthening and improving current and future involvement of the testing team in projects. This means that the testing team may be involved in projects that are related to cyber security, grid and cloud computing, advanced informatics, knowledge technology, wireless technology as well as government-related projects. This entails various technologies and platforms: client-based, web-based, web services and Service Oriented Architecture (SOA), mobile-based, semantic and so forth.
- Test core team – Strengthening and improving the structure of the core team for testing. This is done by establishing a structure that allows for having a group of test specialists and sharing of test knowledge. This is depicted in Figures 9 and 10.

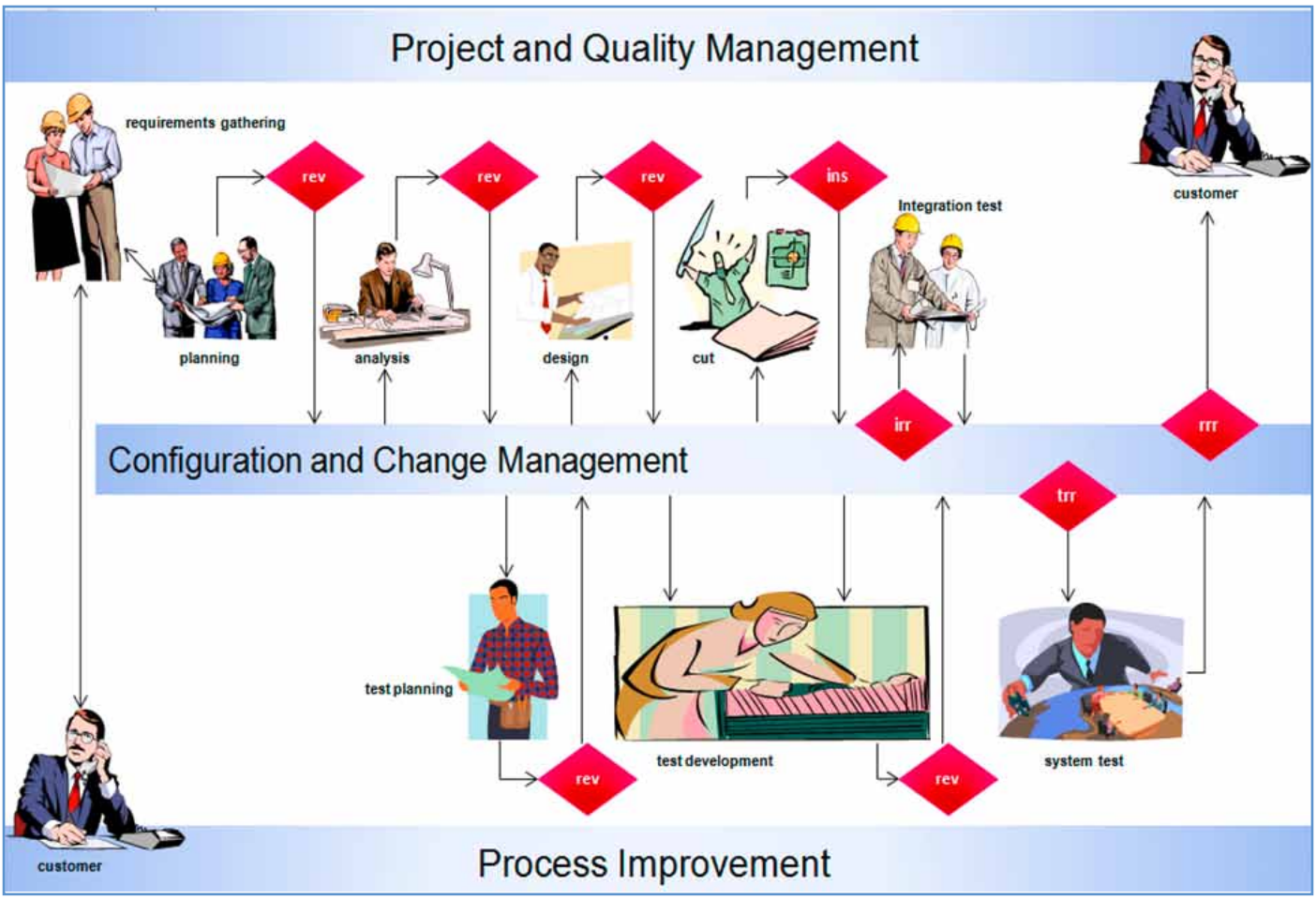


Figure 7: Improvement plan – Test process

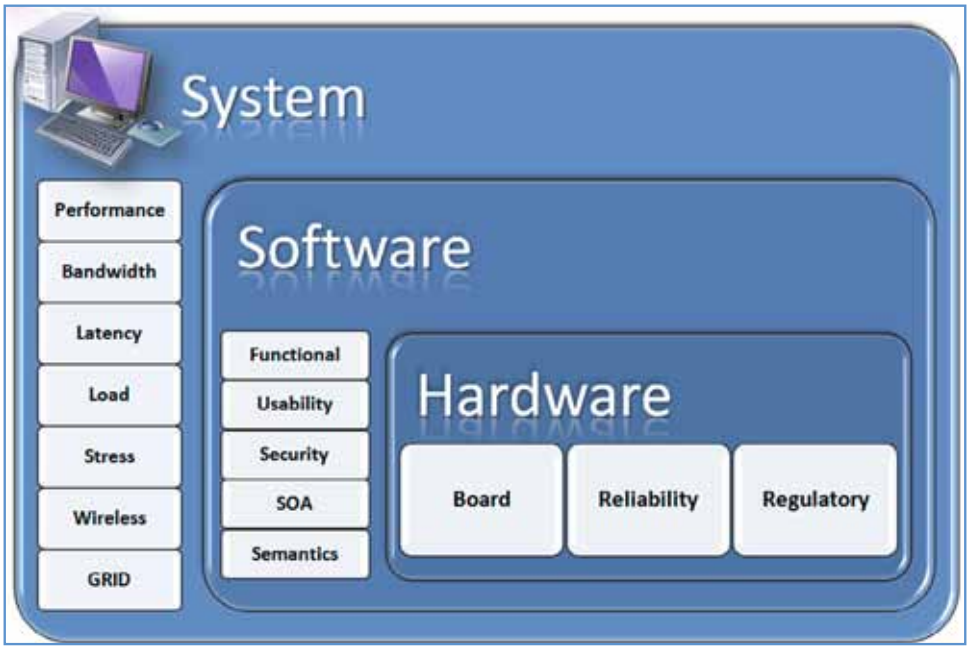


Figure 8: Improvement plan – Test strategies

- Physical test lab – Building the actual lab for testing involvesa server room for keeping the test environment and test tools, working areas for the test engineers, and a usability lab.
- Training and certification – Strengthening and improving the skills and competencies of the test engineersthrough training courses and ensuring they are certified according to their interest and the needs of the organization. These certifications cover but are not limited to, certifications from the International Software Testing Qualification Board (ISTQB®), Quality Assurance Institute (QAI®), Cisco®, Microsoft®, IBM®, Spirent®, EC Council, and Human Factor International.

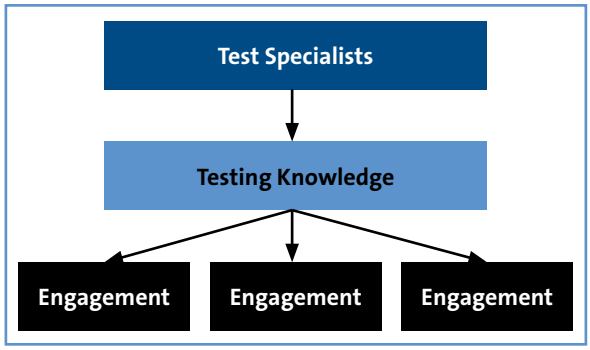


Figure 9: Improvement plan – General structure of the test core team

- Collaboration – Establishing strategic collaboration and partnerships with the players in various areas, such as grid computing, SOA testing, security testing, usability testing and performance testing.
- Test measurement – Strengthening and improving metrics and techniques to measure the test status. At the same time, metrics that are not useful or not applicable are eliminated. Among the metrics collected, analyzed and tracked are the total number of defects, defects by severity, defects open against closed, as well as the number of test cases passed against failed. This is depicted in Figures 9 and 10.
- Test tools – Strengthening and improving the tools to plan, design, execute and report the tests. Earlier tools mostly involved IBM Rational products such as Rational Manual Tester, Rational Functional Tester, Rational Performance Tester, Rational Clear Case, Rational Clear Quest Test Manager and Rational AppScan.
- Automation – Strengthening and improving automation implementation for testing. Among areas put into focus are device testing, network testing, application testing, wireless testing and test reporting. Besides that, automation is extensively done to the infrastructure for the management

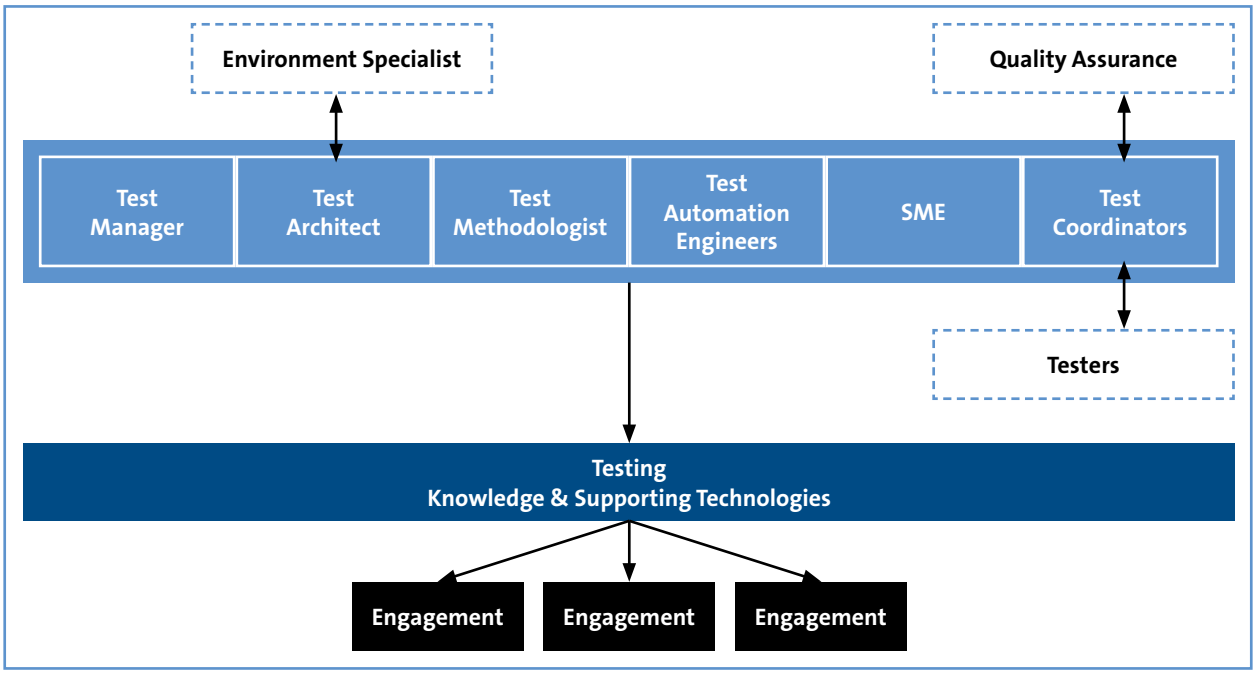


Figure 10: Improvement plan – Detailed structure of the test core team

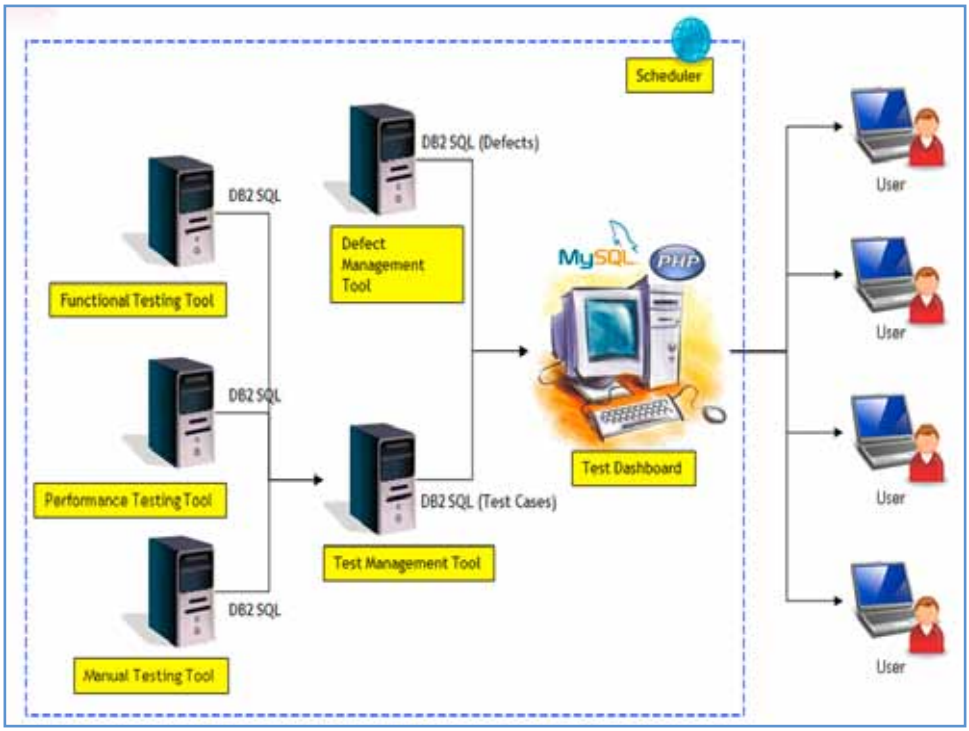


Figure 11: Improvement plan – Automation infrastructure for testing

Figure 12: Improvement plan – Test tactical dashboard

of test execution up to reporting the results to stakeholders. This can be seen in Figure 11.

- Tactical Dashboard – Build a one-stop center for everyone to know the overall status of test progress including test cases results and defects results. This is done via a command center or test tactical dashboard. The skeleton of how the information is presented in the tactical dashboard is shown in Figure 12.

The Way Forward

1. Innovation And Achievement

The journey of setting up the Testing Center of Excellence started in 2006 and since then the department has grown and expanded to a higher level by having more capabilities, competencies and playing bigger role within the organization. To date, various innovations have been introduced to achievethe successful establishment of the Test CoE in the organization.

a. Establishment Of Three Test Labs

Three labs were established:

- System Test Lab – This lab provides areas for tester's workspace, testing showcase, discussion, and most important the test environment center, which is where all test servers are located and managed.
- Usability Lab – This lab is equipped with state-of-the-art eye tracking tools, andconsists of waiting room, observer room, and the usability area.
- Environmental Lab – This lab focuses on hardware testing related to environment and reliability.

b. Introduction of comprehensive testing process and practices

A comprehensive test process was introduced to serve as the main reference for all testing activities in the software development life cycle within the organization. There is also a usability blueprint serving as guidance for developers and designers in developing proper graphical user interface as per industry's best practices.

c. Application of Six Sigma® to predict test defects

As an effort to optimize the testing performance quantitatively, the department has embarked on a Six Sigma® initiative to predict defects in the system testing phase. As aresult, a Test Defect Prediction Model was successfully developed in theform of a mathematical equation using multiple linear regressions to be used for predicting functional defect at the start of the testing phase.

d. Implementation of hybrid tools approach for test execution

Due to the nature of software tested are in mix of platforms, languages and technologies, relying on one standard tool infrastructure is not sufficient. Adoption of various open source tools coupled with existing proprietary tools is heavily implemented for every test strategy due to cost saving as well as better results. Those open source tools are as below:

- Functional test – use of SOAPui, Selenium and QEngine
- Performance test – use of Apache JMeter and OpenSTA
- Network test – use of WANem
- Security test – use of Web Security
- Usability test – use of self-developed Usability Management System
- Test environment – use of Zenoss

e. Research on response time accuracy of different performance testing tools

Due to differences inthe response times given by different performance testing tools during performance testing, whether for load or stress test, a research has been done to investigate the scenario.

ISTQB® Software testing certifications exceed 200,000

Brussels, Belgium, 27 February 2012 – The International Software Testing Qualifications Board (ISTQB®) is proud to announce that, as of December 2011, it has issued more than 200,000 software testing certifications.

Founded in Edinburgh, UK, in 2002, the ISTQB® is a non-profit organisation dedicated to the advancement of the testing profession, and has become the world leader in software testing certification, operating in more than 70 countries through national boards.

From March 2009 to December 2011, the number of certifications doubled, making ISTQB® the most widely adopted and fastest growing software testing certification scheme in the world.

"This is an amazing result," says ISTQB® President Yaron Tsubery, "and demonstrates the value people see in the testing profession and the dedication of its practitioners. With technology playing an ever-increasing role in our lives, it is critical that software intensive products and services are thoroughly tested by skilled professionals."

"I think the scheme has been so successful because we freely offer our syllabi and glossary to the public, which helps to standardise professional terminology. We also offer certifications at a range of levels, from foundation through advanced to expert, allowing testing professionals to be supported right through their careers and keeping them up-to-date with the world's best practices," says ISTQB® Vice President Chris Carter.

"The international nature of the scheme is another strength," says ISTQB® Marketing Working Group Chair Gualtiero Bazzana. "All over the world, people have access to the same syllabi, translated into local languages, which they can learn through self-study or by attending courses delivered by accredited training providers. At the end they sit exams which are externally administered and independently validate their knowledge, and the structure and rules of those exams are consistent throughout the world."

In many ways, the ISTQB® certifications have been accepted as the de facto standard in software testing certification. In addition to certification, ISTQB® also supports innovative research into software testing, runs educational events and conferences, recognises outstanding contributions to the profession, maintains a code of ethics, and shares knowledge.

"At the heart of it is our passion for assisting organizations and professionals to deliver high quality software, and the belief that effective software testing can help make this possible," says Bazzana.

"I would like to thank everybody who has made this achievement possible," says Tsubery, "including ISTQB® volunteers, national boards, training providers, exam providers, and the organizations and individuals who have invested in the ISTQB® Certified Tester scheme. This year marks our 10th anniversary, and we are planning a range of activities to celebrate this and say 'thanks'".

About ISTQB®

The International Software Testing Qualifications Board is a non-profit organisation, headquartered in Brussels, Belgium, whose mission is to advance the software testing profession. It is supported in its operations by 47 member boards, representing over 70 countries.

Further Information:

Gualtiero Bazzana
– ISTQB Marketing Working Group Chair

E-mail: bazzana@istqb.org
Website: http://www.istqb.org

f. Research on Application Life Cycle Management (ALM)

Due to the objective of building and releasing software faster without compromising on quality, the department has embarked on a research initiative on Application Life Cycle Management (ALM) to accommodateboth the existing V-Model as well as the trend towards agile. The initiative incorporated a hybrid approach by using open-source and proprietary tools. The research is not about testing tools only, but includes the entire tools to support application life cycle management, from the requirement inception until end of cycle.

g. Presence and visibility atinternational conferences and symposiums

The department has sent representatives to participate in various international symposiums and conferences via paper presentations or tutorials. These events allow the department the share their actual experiences in setting up a TCoE from scratch and building up its capabilities.

2. Desired Model For Test Center Of Excellence In The Future

Looking at current trends and the demand for high quality software through comprehensive testing, an improved model should be planned and implemented in order toutilize the capabilities of the center to a higher degree. The desired model of the future Test Center of Excellence is presented in Figure 13:

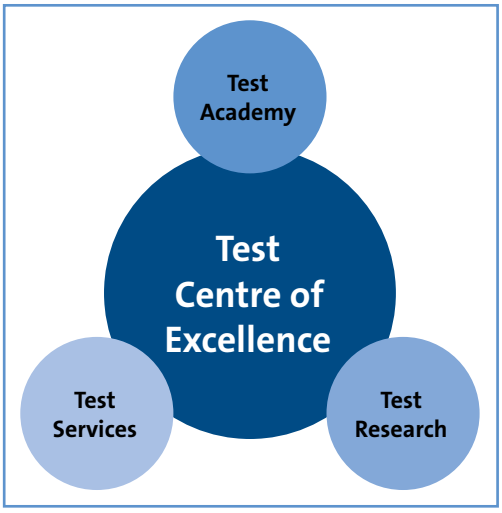


Figure 13: Desired model for Test Center of Excellence

Figure 13 shows three core areas for the future model of the Test Center of Excellence. This is just an initial model, and other areas could be added to this model to suit the context and nature of the organization where testing is being implemented. The description of each area is as follows:

a. Test services

This is a common area for any Test Center of Excellence. The services could be in a form of consultancy or for executing the test. The consultancy services offered can include setting-up of a Test CoE, improving test processes, setting-up a test organization or department, establishing test processes from scratch, selecting suitable tools to execute tests, and other types of consultancy available in the market. As for services for executing the test, these could range from simple functional tests to other non-functional tests, such as performance test, usability test comprising heuristic evaluation and user experience test, security test, network-related tests such as bandwidth and latency test, compatibility test, and SOA testing. Hardware tests such as drop test, water test and vibration test could also be added to the test services.

b. Test research

This might be a new area that should be explored and incorporated into the desired model for the Test Center of Excellence. Such a center should perform extensive research related to software testing as well as other quality-related research. The research is to be more focused on applied research in which the outcome should contribute to test improvement as a whole, such as cost saving, effective automation, introduction of new test tools, process improvement, test infrastructure improvement, and new techniques for testing.

c. Test academy

The idea behind this is to establish collaboration between industry and academia that seriously seestesting as an area that can give significant impact to both parties. Although various engagements have been formed between academia and industry with regard to testing, these have not yet had a lot of impact, neither to the industry in the form of process improvement nor to academia in the form of proper syllabus development on testing. By incorporating a test academy as one of the areas in the future model of a Test Center of Excellence, various initiatives can be formulated and implemented. These include:

- Participation from industry as lecturer or supervisor for undergraduate and postgraduate courses and research atuniversities. This approachcould contribute to providing real world knowledge on testing to the graduating students.
- Placement of students from university in the industry, e.g.,internships for undergraduates,or professional attachment for postgraduates.
- Joint research on software testing between academics and industry practitioners, and preparing a platform for solving real-world problems via the software testing discipline.
- Adoption of industry-centric academic programmes in various forms focusing on testing. For the undergraduate level, a bachelor's degree of software engineering with specialization on software testing could be introduced. For the postgraduate level, universities could offer a master's degree in software testing focusing on comprehensive subjects related to testing, such as functional testing, security testing, usability testing, performance testing, and reliability testing.
- PhD degrees related to software testing could be offered as a new research area by universities with the aim of producing software testing experts to support the increasing demand in this area.

Conclusion

There is no single right model for building a Test Center of Excellence. The model explained in this article is one example from industry that demonstrates how a successful Test Center of Excellence could be established. It is important to identify the key areas that need to be addressed when establishing such a center, so that clear directions, initiatives and strong support by top management can be achieved.This will ensure that a strong Test Center of Excellence is created.

References

3. Black, R. (2002). Managing the Testing Process, 2nd Edition: Practical Tools and Techniques for Managing Hardware and Software Testing. John Wiley & Sons. United States of America.
4. Dustin, E. (2002). Effective Software Testing: 50 Specific Ways to Improve Your Testing. Addison-Wesley Professional. United States of America.
5. Perry, W.E. (2006). Effective Methods for Software Testing, Third Edition. Wiley Publishing. United States of America.
6. CMMI Product Team (2006). CMMI for Development, Version 1.2. Software Engineering Institute, Carnegie Mellon University. United States of America.
7. Materials in Rex Black Consulting Services website (http://www.rbcs-us.com)
8. Materials in SearchSoftwareQuality.com (http://searchsoftwarequality.techtarget.com)
9. Materials in QAI Global Institute (http://www.qaiglobalinstitute.com/Innerpages/Default.asp)
10. Materials in StickyMinds.com (http://www.stickyminds.com//index.asp)

> biography

Muhammad Dhiauddin Mohamed Suffian
is pursuing his PhD (Computer Science) in Software Testing at Universiti Teknologi Malaysia and working as lecturer at the leading open and distance higher learning institution in Malaysia. He was the Solution Test Managerat one of the public-listed IT companies in Malaysia, and prior to that, he was Senior Engineer and Test Team Lead for the test department in one of the leading R&D agencies in Malaysia. He has almost 7 years of experience in thesoftware/system development and software testing / quality assurance fields. With working experience in IT, automotive, banking and research & development companies, he obtained his technical and management skills from various project profiles. A graduate M.Sc. in Real Time Software Engineering from the Center for Advanced Software Engineering (CASE), Universiti Teknologi Malaysia, he holds various professional certifications, namely Certified Six Sigma Green Belt, Certified Tester Foundation Level (CTFL), and Certified Tester Advanced Level – Test Manager (CTAL-TM). He also has vast knowledge in CMMI, test process and methodologies as well as the Software Development Life Cycle (SDLC). He was involved in and has managed various testing strategies for different projects, including functional, performance, security, usability and compatibility test, both at system test and system integration test level. His interestsarethe software engineering and software testing areas, and in particular performance testing and test management.

Redzuan Abdullah
is currently Director of the Product Quality Department in one of the leading R&D agencies in Malaysia. He has more than 14 years of working experience in software development, technical support and software testing in both local and multinational organizations prior to his joining current organization. Redzuan has vast knowledge in various programming languages, CMMI, Six Sigma, telecommunication and project management, and holds various international certifications such asTOGAF, ASP, Red Hat Certified Engineer (RHCE), British Standard Information Security Auditor (BS7799), and Six Sigma Green Belt among others.

Break free!

freedom learning
beyond the classroom

– a unique cloud-based learning solution – simple monthly licensing

- Save 90% on total cost of learning
- On-demand, 24×7 access to global testing training content and resources
- Replicate classroom benefits
- Blend Self Study with Virtual Classrooms and Real Classrooms
- Virtualization with real tutors, real time and real benefits
- Increase certification pass rates
- Value-based licensing – low cost of engagement and implementation
- Scalable, global, secure, private & proven

Visit www.learntesting.com or email enquiry@learntesting.com to find out more about Freedom Learning

Please quote TEXMAG for an additional **10% discoun**t off all Learntesting licenses for Testing Experience Magazine readers

te testing experience

www.learntesting.com

learntesting

Moving Towards a Mature Test Center of Excellence

by Kunal Jadhav

For over 20 years, test professionals have discussed and launched centralized Testing Centers of Excellence (TCoE), yet very few organizations have been able to fully operationalize and industrialize the model. Often, financial services firms implement TCoEs with unclear roles or measurements, making it difficult to accomplish objectives. With our vast experience in setting up Testing Centers of Excellence, we have also experienced demand for hybrid models where financial services QA organizations create a centralized TCoE to optimize processes, and – once formalized – decentralize it over the years to align testing activities with business. While a decentralized testing model can be more business-focused, it often fails to reuse testing assets, standardize frameworks and processes through common tools.

As the testing industry has advanced with more focus on tool-driven testing, there is an increased need for cost and quality control. Leading testing industry tool vendors and open source tool providers offer mature tools to for different types of testing. Today's test professionals have more skills and capabilities than before, contributing experience with tools, practices and methodologie, which helps in finding the right solution for each testing opportunity. With all these factors, it can be difficult to balance the right investments, scale and ROI for testing activities. This is where a Testing Center of Excellence can help.

Why Do You Need a Testing Center of Excellence?

With TCoE implementation organizations can empower their QA teams to achieve time, cost and quality benefits, create and foster an effective knowledge transfer approach, and share best practices across all QA activities. TCoEs can reach these goals by:

- Reducing total cost of testing
- Achieving delivery excellence
- Innovating and industrializing home-grown, open and commercial utilities, tools, and accelerators.

TCoEs provide a structure to standardize testing practices, methodologies, and metrics across all testing projects; they share resources such as processes, templates, tools, tips and estimates; they reduce costs, and they build a direct and robust relationship between QA and business users. Outsourcing services further can provide benefits by combining lower costs with mature and proven testing processes, which helps jump-start your TCoE.

What Makes Up a Testing Center of Excellence?

When well executed, a TCoE can help your organization scale testing services to meet peaks and valleys in QA activities, and to offer new or specialized services and skills to improve schedules and quality. To achieve these benefits, your centralized TCoE must be planned strategically and include the following elements.

Strategic enablers

TCoEs are created to support a long-term vision for testing across your financial institution. This vision requires a roadmap which is periodically reviewed and evaluated, and a method for monitoring the ROI from various tools, accelerators, and processes that make up your investments in the TCoE. A good foundation for the TCoE is to align with the business and partner with leading tool providers. This helps the TCoE increase capabilities across the different skill sets, tools and accelerators that will be required.

Process and methodology

Every TCoE setup initiative provides a robust estimation approach for all testing types and methodologies. These building blocks help delivery teams create standard and repeatable project estimates, and provide well-defined structured test methodologies which outline the scope of testing for projects aligned to the TCoE setup.

Infrastructure

TCoE setup also requires well defined infrastructure support including a test lab equipped with key testing tools, utilities, virtualization and cloud-based environments to test across various platforms. Infrastructure is a critical component to meet TCoE objectives across platforms, servers, tools, and technologies.

Potsdam

The most beautiful place for bright minds.

The Schiffbauergasse, located in the creative heart of Potsdam, is one of the many locations where innovative work is being carried out today. This lovely waterfront site was used in various ways in the past – for the building of steamships, the production of a coffee substitute, fish-breeding, and even for drilling Hussars! Here one finds a surprising mix on spacious and airy twelve hectares. Right on the water's edge, a vibrant art and cultural scene meets high tech companies, as a fascinating history merges into a pioneering future.

Come to Potsdam and discover the city for yourself. Experience this baroque city as a lovely holiday destination, as a multi-faceted event location, or even as an attractive location for your business.

Potsdam: a great place for great ideas.
This magnificent city welcomes you warmly!

POTSDAM

State Capital Potsdam
Department of Business Development
economy@potsdam.de I www.potsdam.de

Organization

Managing and running a TCoE requires a strong organizational structure to handle knowledge management, people skill development, demand and supply management, communication with internal and external customers, and performance management. A strong organizational policy can foster highly motivated and efficient TCoE teams.

These components illustrate the investment required to set up a TCoE which meets enterprise-wide requirements.

The Testing Center of Excellence Journey

For any organization considering a Testing Center of Excellence model, the first step is to assess the current state of their testing organization. This assessment lets a company gauge the maturity and competitive positioning of current test processes and understand the steps required to achieve business-specific goals by aligning test processes with business imperatives. Once the assessment is complete, setting up a TCoE requires:

- Well-defined, measurable objectives: What are our goals?
- A clear vision and roadmap for setting up the TCoE: What are the key objectives to set up the TCoE?
- A definition of the TCoE roles and responsibilities: Who is responsible for what?
- Quality and productivity benchmarks with industry trends: How will our testing efforts compare against our competitors and industry?
- Resource transition or acquisition plan: How do we get the skilled resources we need?
- Innovative and efficient operating models: How will we work?
- TCoE execution plan: How do we reach our goals?
- Monitoring and control framework: How do we measure our goals?

TCoEs can bring enormous benefits to QA organizations due to their strategic nature. If TCoE efforts are not strategic, the journey to a TCoE model can result in lost credibility, frustrated customers and a failed initiative that has squandered the firm's investments and expertise. Many financial organizations have successfully launched TCoEs by partnering with third party vendors to utilize proven processes and roadmaps. Outsourcing the creation of a TCoE to a vendor can also result in lower costs through the use of offshore skills and resources.

At Capgemini we use a three phase approach to set up TCoEs: Transition, Stabilization and Optimization.

Transition

The Transition phase involves moving from the existing operating model to a defined target operating model. Change management plays a vital role during this phase since there are often changes in the testing organizational structure, reporting, and other processes. These changes should take place with no or minimal impact on end customers, so firms must consider projects in flight and release schedules when planning the transition.

If TCoE set-up is being outsourced to a testing service provider, then knowledge transition and resource loading will also play a vital role. Depending on the size of your organization and locations for operations, the transition phase usually takes six to nine months.

Stabilization

The core objective of the Stabilization phase is to operate smoothly with the new operating model and processes. The TCoE will start gathering SLA-driven measurements and dashboards such as Goal-based Quality Metrics to quantify the progress and benefits of all activities. This continuous measurement approach results in benchmarking data across projects.

Optimization

In the Optimization phase, the TCoE starts delivering as per defined SLAs and looks for opportunities to improve processes across the entire testing life cycle. This phase starts with the definition of new targets for a predetermined period; for example, test case efficiency LCL = 98.5%, defect leakage to next phase = less than 2%. The periodic dashboard and metrics collection is coupled with variance calculations against the defined targets to measure progress. These new targets encourage innovation and foster the creation of new accelerators.

While building a TCoE, the time from Transition to Stabilization to Optimization should be monitored to make sure the planned roadmap is being followed and the initiative hasn't lost focus. Based on our experience within two or three years, most financial services institutions can execute a TCoE initiative that fosters a TCoE culture and realizes a return on investments.

Overcoming Challenges

Setting up a Testing Center of Excellence can also involve several challenges:

- Current testing teams may not want to see the testing function centralized
- Challenges to implement governance and control frameworks for the TCoE set-up
- Lack of clear TCoE objectives and measurement techniques
- Perception of 'body shop' model and process overhauls prevents support for the TCoE
- TCoEs are often executed in a pooling model, which may increase overhead costs, and lead to budget restrictions

By following a structured approach when implementing a TCoE model, these risks can be mitigated through a set of activities such as clearly defining measurable objectives and utilizing change management practices; both are part of the Transition phase.

Conclusion

The TCoE approach continues to evolve, and at Capgemini we have seen huge transformations in testing organizations at financial services firms. Many organizations are exploring centralized, decentralized or hybrid operational models for testing in addition to outcome-based TCoE models. The financial services industry is a dynamic, ever-changing marketplace, and we have seen a consolidation among global testing services vendors to meet the needs of this industry. When using an outsourcing model, financial institutions are not willing to lose control over time,

cost and quality, so they are looking for innovative approaches for their TCoEs.

For this reason, a Testing Center of Excellence should be loosely coupled within the organization with dual reporting to both the TCoE structure and delivery to ensure the TCoE is accountable to delivery. Such TCoEs help standardize testing practices and metrics across all testing projects, reduce costs and achieve a more direct relationship between QA and the business.

> biography

Kunal Jadhav
is a Senior Manager with the Capgemini Financial Services business unit, leading the Testing Center of Excellence in India, focusing on automation, performance, security and mobility testing as well as on test processes and consulting. In this role, Kunal is responsible for building testing solutions and capabilities for financial services clients to foster the reuse of assets and accelerators across customer projects to achieve productivity and quality improvements. Previously, also at Capgemini, Kunal managed two large-scale testing engagements for tier 1 global banks leading teams of over 400 test professionals across three regions. He has also led testing maturity assessments for large IT organizations using industry standard assessment frameworks.

The Future of Test Centers: A New Vision

by Kaspar van Dam & Tjerko van der Wouden

Being fair about testing

Nowadays testing is more than the demonstration of quality. It's also more than searching for bugs. It is even more than checking if the developed software is actually the product which was requested by the business. Testing has become a total package which accompanies the acceptance of an application throughout its entire life cycle. This comes at a time when organizations are facing the prospect of cutting costs and shortening the time-to-market, whilst at the same time improving quality. Organizations also face the fact that ICT products are more and more integrated with their businesses and tend to get more complex and larger in scale. This can result in testing consuming well over 40% of the budget of an ICT project.

However, not only the world of testing is changing, but there are also changes within the management of projects. Some organizations are replacing linear, Waterfall projects for agile (or SCRUM) projects, while other organizations continue to profit from linear or Waterfall projects. Test organizations often make a definite choice: either one OR the other. However, while developing a Test Center, we have chosen for the best of both worlds and use agile techniques within linear projects and linear techniques for agile projects.

At Ordina, we found it was time for innovation. A new vision on testing, without reinventing the wheel. Re-using existing standards, but with another look on testing. Only test what really needs to be tested and use the strength of the developer of an application. Simplify test processes and uniform the way of reporting test results. In short: Ordina believes in the centralization of the testing organization and in keeping a focus on efficiency instead of just a focus on quality. However, it is needless to say that within the Ordina Test Center quality will remain a priority as well.

For an organization it is important to achieve its goals. The Ordina Test Center will provide methods, techniques, processes and standards. The only thing that remains for an organization is the result. That's the new form of testing according to the Ordina Test Center.

How the Ordina Test Center works

The Ordina Test Center consists of an integral team of test professionals. The members of this team work according to standards based on TMap, supplemented with best practices from Ordina. These test professionals can be put to work at a client location or at the Ordina offices in Groningen or Nieuwegein (The Netherlands).

We start with the test base. This is where we determine what needs to be tested. Testing the right thing, that's what matters. The consultants of Ordina create a test base based on existing documentation, supplemented with input from stakeholders. We call this 'testbase 2.0'. Using an efficient and standardized test process, we will help execute the right tests or execute these tests all together. During execution, findings are captured and reported in a uniform way giving insight in what has been made.

As mentioned, within the Ordina Test Center our focus is on efficiency and on achieving the goals of an organization. While traditionally testing is positioned as a separate part of the organization with a focus on quality and is considered a key component for accepting or not accepting an ICT product by the organization. When looking at the V-model, we see design and development on the left side of the model and testing on the right side. We see this both with Waterfall and with agile projects, the difference being that with agile projects the V-model will be used over and over again for every sprint. Especially with current developments around agile, the role of testing has changed, and will keep changing. Within a project everyone is a team member, the strong dividing line between design, development and testing will fade. A project just needs sufficient expertise about testing on board. At Ordina we feel that this development should also be adopted within linear Waterfall projects. That is why we put the Test Center in the middle of the V-model, instead of the traditional position of testing on the right side (see figure 1). Creating an ICT-product becomes a process between ICT and Business again, where testing only supports the process of creating a product (ICT) that will be accepted by the end user (Business). Business and ICT are together responsible for a good result with the right quality. The Ordina Test Center will tell both parties what to test to achieve the desired goals that were described within the 'testbase 2.0'.

The ASQF Career Center

We‘ve got the right job for you.

Consultants (m/f)
KUGLER MAAG CIE GmbH, Kornwestheim

Consultant & Trainer (m/f)
SOPHIST GmbH, Nürnberg

Microsoft TFS Specialist
CMCS GmbH, Erlangen/Nürnberg

Consultant SW Quality Management (m/f)
Software Quality Lab GmbH, München, Wien oder Linz

Manager Tool Evaluation Center (m/f)
Software Quality Lab GmbH, München, Wien oder Linz

Software Test Engineer (m/f)
Software Quality Lab GmbH, München, Wien oder Linz

Software Tester (m/f) SOGETI Deutschland GmbH, Düsseldorf, Frankfurt, München, Hamburg und Stuttgart

Software Tester – Load & Performance (m/f) SOGETI Deutschland GmbH, Düsseldorf, Frankfurt, München, Hamburg und Stuttgart

Software Tester – Testmanager (m/f) SOGETI Deutschland GmbH, Düsseldorf, Frankfurt, München, Hamburg und Stuttgart

Software Tester – Test Automation (m/f), SOGETI Deutschland GmbH, Düsseldorf, Frankfurt, München, Hamburg und Stuttgart

Consultant/System Engineer Infrastructure Services (m/f)
SOGETI Deutschland GmbH, Hamburg

Test Manager (m/f) with Banking Know How ISMC Information System Management & Consulting GmbH, München, Frankfurt oder Karlsruhe

Software Tester (m/f) with Banking Know How ISMC Information System Management & Consulting GmbH, Waldbronn

Software Developer .NET (m/f)
GFB Softwareentwicklungsgesellschaft mbH – Q-up, Oberursel

Software Tester (m/f) for Quality Assurance
GFB Softwareentwicklungsgesellschaft mbH – Q-up, Oberursel

Consultant Test Data Management
GFB Softwareentwicklungsgesellschaft mbH – Q-up, Oberursel

Test Manager (m/f)
R+V Allgemeine Versicherung AG, Wiesbaden

Software Developer (m/f)
GFB EDV Consulting und Services GmbH, Oberursel

Departmental Manager for Telecommunikations (m/f)
IABG mbH, Ottobrunn

Consultant Digital BOS Radiocommunication (m/f)
IABG mbH, Ottobrunn und Berlin

Senior Consultants IT Management & Quality Services
Díaz & Hilterscheid Unternehmensberatung GmbH, Berlin

Consultant – Specialist for SCM, ALM and Test Automation (m/f) elego Software Solutions GmbH, Berlin und Dresden

Basic-Software Developer LIN (m/f)
Brose Fahrzeugteile GmbH & Co. KG, Coburg

Software Developer (m/f) AUTOSAR
Brose Fahrzeugteile GmbH & Co. KG, Coburg

Examination Assessor (Certified Agile Tester ®) (m/f)
iSQI GmbH, global

Invigilator Examination (m/f)
iSQI GmbH, München, Stuttgart, Konstanz

Examination Assessor / Office Management (m/f)
iSQI GmbH, Potsdam

Software Architect (m/f)
infoteam Software AG, Dortmund, Erlangen und Zürich

Software Developer Java (m/f)
infoteam Software AG, Dortmund, Erlangen und Zürich

Software Developer C#, .NET (m/f)
infoteam Software AG, Dortmund, Erlangen und Zürich

Quality Assurance Engineer / Software Tester Test Automation (m/f) Sedo GmbH, Köln

Software Developer (m/f)
ERGO Direkt Versicherungen, Nürnberg

Software Tester (m/f)
ckc ag, Braunschweig

Trainee Test Specialist / Junior Consultant Test (m/f)
ckc ag, Braunschweig

Lead Testmanager at Healthcare Syngo (m/f)
Siemens AG, Erlangen

Consultant and Trainer (m/f)
SOPHIST GmbH , Nürnberg

IT-Consultant (m/f)
modulo3 GmbH, Nürnberg/Region Oberfranken

Software / System Test Engineer (m/f)
Valyue Consulting GmbH, Stuttgart

Detailed information on all job offers on www.asqf.de/stellenangebote.html *(German only)*

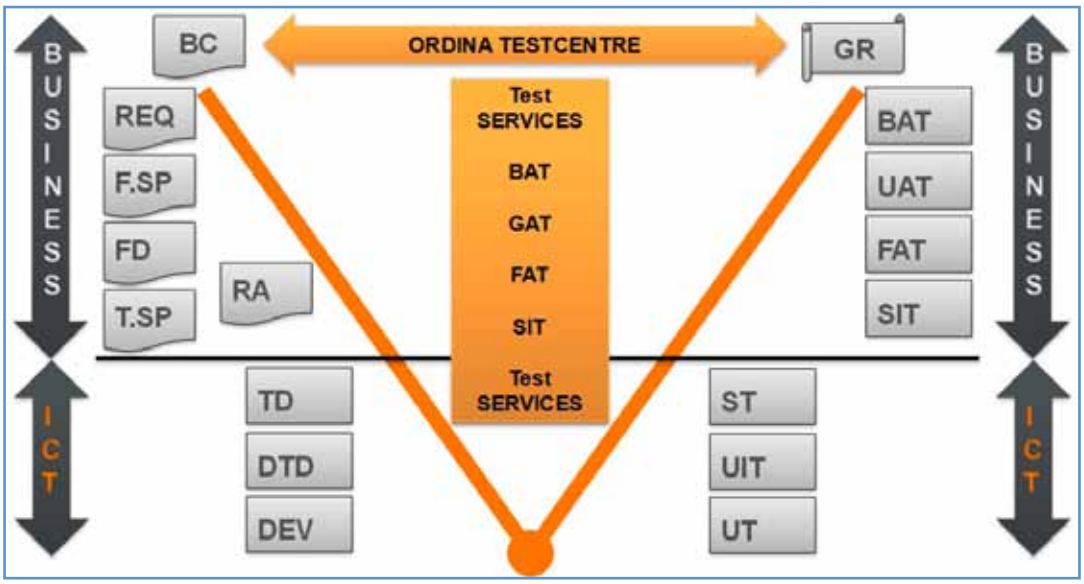


Figure 1: Ordina Test center positioned within the V-model

While doing so, we will always keep a focus on efficiency. When we can achieve the same goals with less effort, then we will do so. This will be done by standardization and centralization, by an efficient translation of the 'testbase 2.0' to tests, and, when possible, also by automating tests using different testing tools. Especially within agile projects, repetitive tests lend themselves for automation of (regression) tests, but this could also be of interest when working with different releases within a Waterfall project. Within the Test Center we will provide the necessary know-how of automating tests and can advise organizations about test automation. This means that test automation can be adopted by an organization with a relatively small investment in know-how, enabling the organization to start with test automation with a head start.

Test center as a team member in agile and Waterfall projects

As pointed out earlier, a lot of developments are visible within the testing world. A big challenge for any testing excellence center is the emergence of agile projects. The role of the tester no longer exists, there are only team members that need close interaction with other team members in order to reach a common goal: an accepted ICT product. It is the vision of the Ordina Test Center that both with Waterfall and agile projects the Test Center can take the role of a team member, instead of a separate test organization. By positioning the testing expertise in the center of the V-model, we also place testing in the center of any project. To create a 'testbase 2.0', we interact with business and IT at the left side of the V-model. For specification and execution of tests, we interact with both sides of the model. The business can tell us what to test, IT tells us how to test, and we interact with acceptors to help the organization in accepting the developed IT product(s). Business and IT are again owner of the validation process, testing only supports this process. Testing itself is no longer the driving force behind the (non-)acceptance of the result.

Sourcing

With the Ordina Test Center it is possible to hire testers, whereby co-sourcing is also a possibility. In this case, testers will be hired in a smart way, with a higher form of integration on planning and forecasting. When working on a project base, a test partner can also be hired based on a project contract: When an organization prefers to stay in control while Ordina accepts test assignments on a fixed price basis, then it is possible to choose for out-tasking. The most intimate form of integration, however, is outsourcing, where the Ordina Test Center will actually become part of an organization. Arrangements will be made on KPIs and cost reduction, and the Test Center will offer its services at fixed price and fixed date. In this case all testing efforts will be in the hands of the Ordina Test Center.

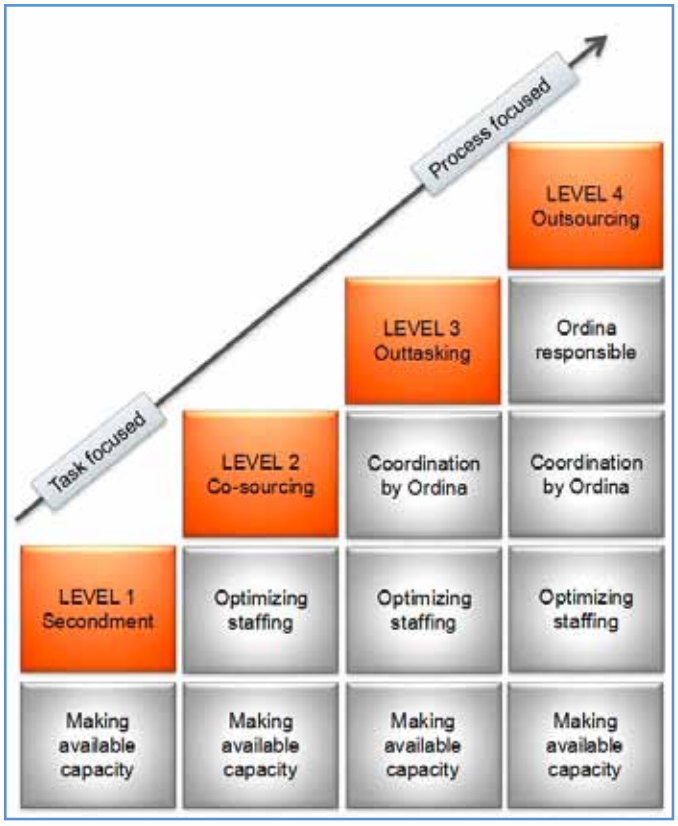


Figure 2: Contract forms

Whichever form of sourcing an organization chooses: the Test Center will always use the same process with the same standards. Financial benefits will increase when staffing is more optimized and coordination is more in the hands of Ordina. When an organization decides to make the Ordina Test Center responsible for the result, arrangements can be made concerning KPIs and maximum costs to assure the result for the organization. That's being fair about testing.

> biography

Kaspar van Dam
has been active in the testing field since 2005 and has specialized in test automation and performance testing since 2009. During the last years he has had different roles varying from test tool engineer to test consultant at a number of companies. His experience covers both the technical implementation of test automation and performance testing and the different management and coordination tasks surrounding this work area. He is also involved in the development of testing in general within his company (Ordina, Netherlands) with a focus on test automation. Currently, he is working in the field of product acceptance testing at a large Dutch company within the financial market.
At Ordina he has taken part in the setting up of a Test Center with a special focus on test automation. Together with project leader, Tjerko van der Wouden, and a number of colleagues, a new vision was created around the phenomenon of 'Testing Centers of Excellence', in which testing is no longer a separate division within a project, but a more central part within the process, starting from design and ending with acceptance of the software .

A SPICE compatible Process Assessment Model for TPI®

by Sven Koos

A SPICE[1] compatible Process Assessment Model for TPI®[2]

The implementation of an assessment is considered in the testing community as an appropriate measure for determining the maturity of the software development process of an organization. The choice of a suitable **Process Assessment Model (PAM)** is important in relation to the presentation of the assessment results and the specifications for a subsequent process improvement project.

This goal also applies to a dedicated consideration of the testing process: With an assessment the maturity of the testing process in terms of a concrete development project can be measured, and the compliance of the implemented test process with an established standard can be determined. Specific proposals to improve the testing process can be developed; and an improvement project can be initiated.

1. The Development Organization

The fundamental criterion for the selection of an appropriate PAM is the intended focus of the assessment: Will a single project, a set of projects (project focus), or the entire development organization of a company (organizational focus) be assessed?

PAMs with project focus determine the lower maturity levels exclusively with project-related indicators and consider the integration of a project into the entire development organization only at the upper maturity levels. They are suitable for both small and medium-sized, as well as for larger companies.

SPICE (ISO / IEC 15504) has been developed as one of the most successful PAMs with project focus, is internationally standardized (IS 2006) and forms the basis for some major industry-specific assessment models (e.g. Automotive SPICE).

1 *SPICE – Software Process Improvement Capability Determination, ISO/ IEC 15504 IS 2006*

2 *TPI® – Test Process Improvement, Registered Trademark of sogeti*

2. The Scope of the Assessment

Test and verification are two basic methods of quality assurance that perform important functions in the software development cycle. In contrast to this importance, there is a partially incomplete or superficial representation of the testing process in many PAMs, including SPICE. Specific improvements of the testing process are not offered by the models.

To overcome the lack of specificity, some models for the assessment and improvement of the test process in its entirety and complexity were developed. TPI is one of the accepted and proven models in this area with a clear focus on improving the test process in a project environment.

However, TPI does not fulfill the requirements for a standardized PAM, such as those formulated in SPICE. Content of this article is the representation of a SPICE-compatible PAM for TPI and its development and application in practice.

3. An applicable Process Assessment Model

The basic motivation for the development of this PAM was:

The PAM should standardize the maturity determination of the testing process and make the estimation transparent, comparable and understandable for testing professionals who are not familiar with process assessments. For this reason the proven measurement framework of SPICE, which is used in many assessment models, is applied.

In this PAM, the content of the TPI model should be converted and maintained with all its elements.

The PAM fulfills the requirements of ISO / IEC 15504 in this structure (Figure 1).

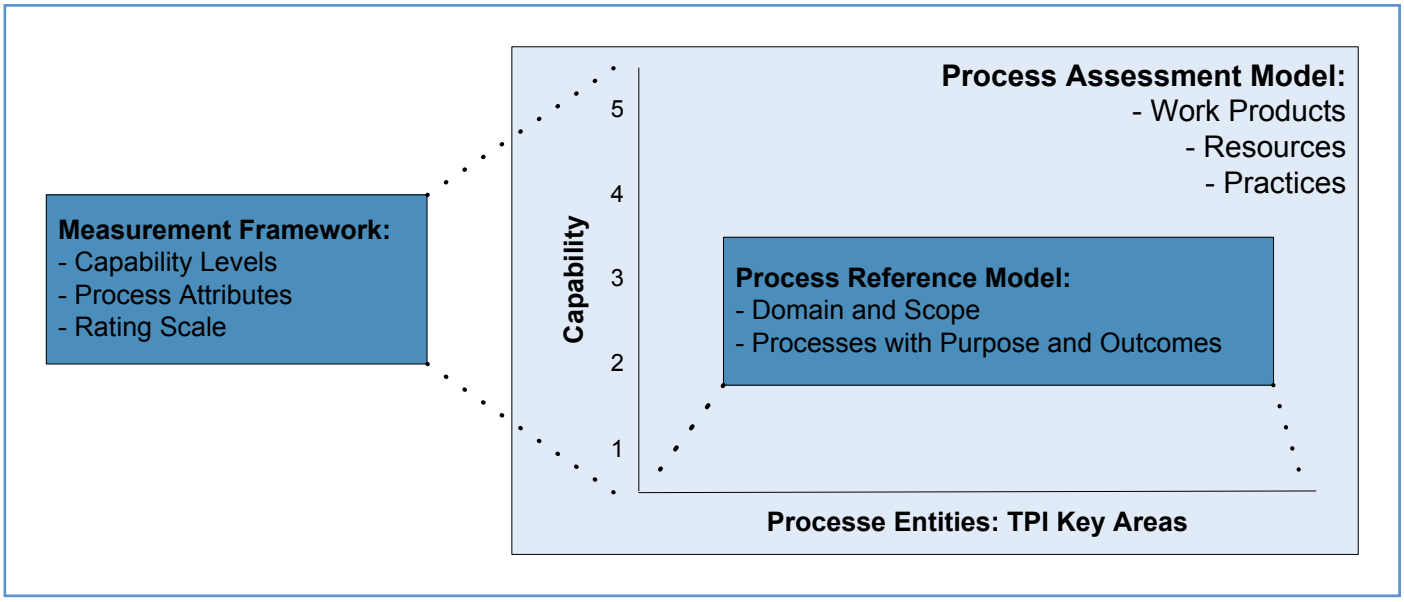


Figure 1: Structure of the Process Assessment Model (PAM)

3.1. Measurement Framework – The Capability Dimension

In the assessment model, the maturity of the individual processes is expressed in terms of **Process Attributes (PAs)**, which are summarized in the following capability levels (only the industrially most relevant levels 0-3 are listed here):

Level 0 (Incomplete process): The process is not implemented, or fails to achieve its process purpose. At this level, there is little or no evidence of any systematic achievement of the process purpose.

Level 1 (Performed process): The implemented process achieves its process purpose.

Level 2 (Managed process): The performed process is now implemented in a managed fashion (planned, monitored and adjusted) and its work products are appropriately established, controlled and maintained.

Level 3 (Established process): The managed process is now implemented using a defined process that is capable of achieving its process outcomes.

PAs are characteristics of a process, which on the basis of a rating scale can be estimated and thus provide a measure of the fulfillment of the process. Each attribute measures a certain aspect of the process capability. In the developed PAM the measurement of process capability is based on nine PAs that are defined in ISO / IEC 15504-2 (only attributes of the industrially most relevant levels 1-3 are listed here):

Level 1
PA 1.1 Process performance: measure of the extent to which the process is performed

Level 2
PA 2.1 Performance management: measure of the extent to which the performance of the process is managed

PA 2.2 Work product management: measure of the extent to which the work products produced by the process are appropriately managed

Level 3
PA 3.1 Process definition: measure of the extent to which a standard process is maintained to support the deployment of the defined process

PA 3.2 Process deployment: measure of the extent to which the standard process is effectively deployed as a defined process to achieve its process outcomes

The PAs are rated on a four-step rating scale, as defined in ISO / IEC 15504-2.

Not achieved (N): 0-15%
Partially achieved (P): 16-50%
Largely achieved (L): 51-85%
Fully achieved (F): 86-100%

In practice, this graduated rating scale leads to a more realistic, accurate and meaningful picture of the existing capability level of the process than, for instance, using the TPI model with a two-step rating (achieved, not achieved).

The achievement of PAs is shown on the basis of evidences relating to the defined assessment indicators. There are two types of assessment indicators:

Process Performance Indicators (only for capability level 1):
Base Practice (BP),
Work Product (WP).

Process Capability Indicators (for capability levels 1 to 5):
Generic Practice (GP),
Generic Resource (GR).

BPs determine the activities for achieving of the process purpose and process outcomes. WPs define the upstream and downstream work products of the process.

For each process the indicators for the process performance (BPs, WPs) are defined in the process dimension (see chapter 4.2). Those indicators are used to measure the degree of implementation of the process performance attributes (capability level 1), e.g.

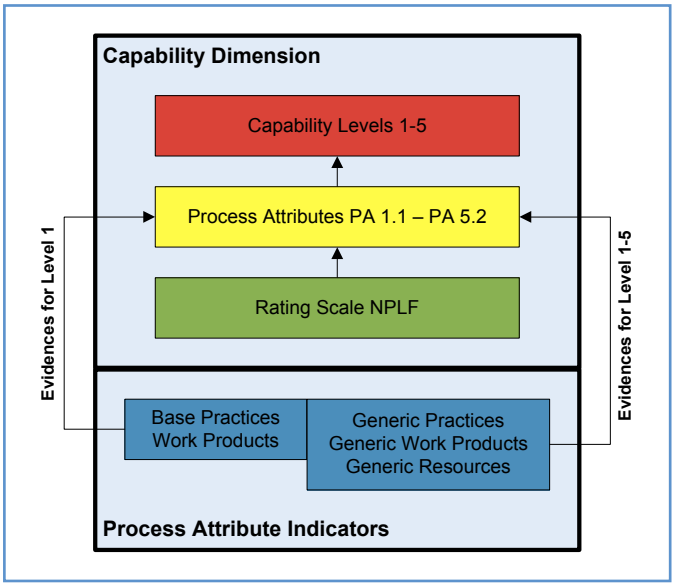


Figure 2: Structure of the Capability Dimension

Process: *TPI.5 Test specification techniques*

Base Practice: *TPI.5.BP1:*
"Specify test cases by means of a described technique"

Work Product: *Test case specification*

The indicators of the process capability (GP, GR) give important information about the extent of implementation of the attribute in the running process and involve significant activities and resources (e.g. tools, methods, results databases, etc.) associated with the implementation of the attribute's purpose in the process, e.g.

Process Attribute:
PA 2.1 Performance management

Generic Practice:
GP 2.1.2 "Plan and monitor the performance of the process to fulfill the identified objectives"

Generic Resource:
Project planning, management and control tools, including time and cost reporting

Figure 2 shows the overall structure of the capability dimension.

3.2. Process Reference Model – The Process Dimension

3.2.1. The TPI Model

The TPI model, with a set of checkpoints in all areas of the testing process, offers the ability to identify the strengths and weaknesses of an implemented test process and to make dedicated suggestions for improvement.

The TPI model includes 20 key areas on the following topics:
Test Phases, e.g. „Life-cycle model" and „Moment of involvement",
Test Techniques, e.g. „Static test techniques" and „Metrics",
Infrastructure, e.g. „Test environment" and „Office environment",
Organization, e.g. „Communication" and „Testware management".

The key areas are focused on high-level testing (e.g. system and acceptance testing). For low-level tests (e.g. module test) and evaluation, separate areas have been defined.

The TPI model also includes levels and development stages, which are considered important when later combining the established measurement framework and defined **Process Reference Model (PRM)** into a complete PAM.

To ensure a comparable estimation of a key area, each level defined in the model is supplied with checkpoints that must be met to achieve this level, e.g.

Key Area:1. Test Strategy

Level: *A*

Checkpoint 1.A.1:
"A motivated consideration of the product risks takes place, for which knowledge of the system, its use and its operational management is essential."

In order to reach the checkpoints of the individual key areas, a number of optimization proposals will be made to support the process improvement.

The representation of the interrelationships and interdependencies of key areas, levels and development stages is illustrated in the TPI Test Maturity Matrix. Figure 3 shows the general elements of the TPI model.

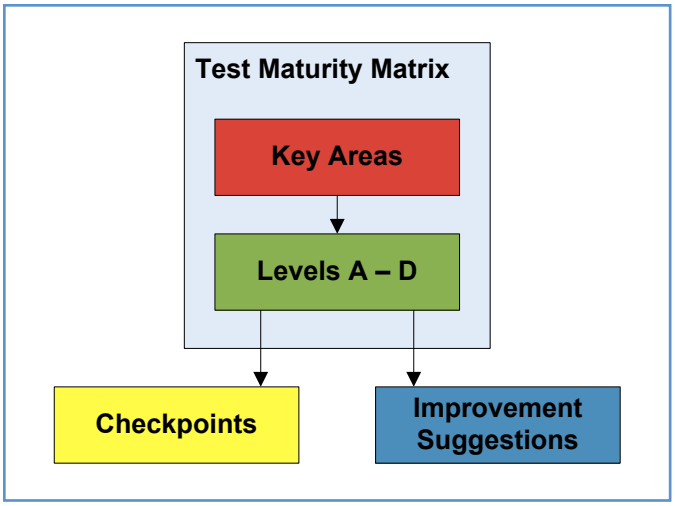


Figure 3: Elements of the TPI Model

3.2.2. Process Entities

For the creation of the PRM, the key areas of the TPI model are equated to processes. The process TPI.21 Integration Testing as an extension has already been implemented by Sogeti in the model TPI Automotive ([2]) and reflects the increasing importance of the integration of software components and their interfaces tests.

3.2.3. Process Purpose and Outcomes

The transfer of the TPI key areas into the PRM meets the following requirements of the SPICE standard for an appropriate model by derivation from descriptions in the TPI model:

S o f t w a r e T e s t e r

Curso Certified Tester

ISTQB® Certified Tester – Foundation Level
Madrid - Barcelona- Sevilla - En Castellano

OBJETIVOS
El seminario **«Certified Tester - Foundation Level»** le proporciona los procedimientos y métodos más importantes de prueba con los que podrá preparar y realizar pruebas de software de forma eficiente y eficaz. Una vez completado el seminario podrá aplicar las técnicas aprendidas en sus proyectos y así contribuir decisivamente al éxito conjunto de los mismos.

CONTENIDO
El seminario se orienta según el plan de estudios de la International Software Testing Qualifications Board.
Se abordarán los siguientes temas:
- Fundamentos de las pruebas de software
- Pruebas durante el ciclo de vida del software
- Procedimientos de prueba estática
- Procedimientos de prueba dinámica
- Procedimiento del diseño de pruebas
- Gestión de pruebas
- Herramientas de prueba

PRESTACIONES
El precio del seminario incluye:
- Realización del cursillo con un máximo de 14 participantes
- Almuerzos y Coffee Breaks
- Un ejemplar impreso de la documentación del cursillo

EXAMEN OFICIAL
A continuación del seminario si se desea puede realizarse el examen de «Certified Tester - Foundation Level» conforme a ISTQB. ¡98% exámenes aprobados en España!

REQUISITOS
Experiencia práctica en proyectos informáticos, conocimientos básicos de desarrollo de software.
La experiencia en la realización de pruebas de software es útil, pero no es requisito imprescindible para participar.

AUDIENCIA
Software Tester
Director de proyecto
Gestor de calidad
Probador de software
Desarrollador de software

ACCREDITED ISTQB Certified Tester Foundation Level TRAINING PROVIDER

¡Más de 200,000 testers certificados en el mundo!
+
99% Aprobados en España con **nexoQA**

Descuentos para los asistentes de expo:QA'12

Trainings expo:QA

Organizado por: nexo qa

Acreditado por: ISTQB International Software Testing Qualifications Board

Más Información y Registro:
www.expoqa.com

Definition of the process purpose and the outcomes of the process performance,

Determination of the roles involved in process execution.

3.2.4. Base Practices and Work Products

The indicators for the PA of capability level 1 (BPs, WPs) are formulated on the basis of the TPI checkpoints.

The BPs and WPs are associated with the outcomes of process performance. Since BPs, WPs (out of TPI checkpoints), and process outcomes (out of TPI process descriptions) are derived from various elements of the TPI model, there is no 1:1 correspondence between practices and outcomes as in other reference models based on SPICE. In practice, this is not a significant limitation of the model, since usually an estimation of the process performance is done directly on the basis of the evidenced BPs and WPs. In the PRM those are more extensively described than the outcomes of process performance.

The WPs will be defined according to the determined BPs and resulting from the implementation of the process. The explicit naming of WPs as a precondition for the implementation of the process is omitted because of the limitation of the PRM to the area of Test & Verification.

3.2.5. Improvement Suggestions

The description of improvement suggestions is a specific feature of the TPI model. This is very helpful for the content definition of projects for process improvement. Therefore, all process-related recommendations and suggestions for improvement are incorporated into the new PRM unchanged.

Figure 4 shows a process example of the PRM.

Process
TPI.12 Test functions and training

Process Purpose
The purpose of the Test functions and training process is to ensure the correct composition of a test team and a mix of the various disciplines, functions, knowledge, and skills.

Process Outcomes
As a result of successful implementation of this process:
[Outcome 1] A good composition of the test team with different skilled testers such as users and developers is given.
[Outcome 2] Testers have specific knowledge and psychological skills.

Process Priority
Priority 3

Improvement Suggestions
Give testers and test managers test training.
Include function descriptions in the test plan, which describe who should perform which task.
Arrange the required subject matter expertise in a timely manner.

Base practices
TPI.12.BP1: Set up at least a test manager and several testers. [Outcome 1]
TPI.12.BP2: Define tasks and responsibilities. [Outcome 1]
TPI.12.BP3: Provide specific test training. [Outcome 2]
TPI.12.BP4: Make expertise in the subject matter available. [Outcome 2]

Output work products
Training material [Outcome 2]
Human resource management plan [Outcome 1]
Project plan [Outcome 1]
Software test plan [Outcome 1]
Validation test plan [Outcome 1]
Training plan [Outcome 2]
Training record [Outcome 2]
Human resource needs analysis [Outcome 1]
Knowledge management strategy [Outcome 2]
Training strategy [Outcome 2]

Figure 4: Example process of the Process Reference Model (PRM) (TPI.12 Test functions and training)

3.3. Assignment of TPI Levels and SPICE Capability Levels

The merging of the established measurement framework and the defined Process Reference Model (PRM) to a complete Process Assessment Model (PAM) takes place with the assignment of TPI levels A through D to the SPICE capability levels 1 through 5 for each process.

It is noted that not all TPI key areas completely consist of the levels A through D, and the levels in several key areas are distributed in the TPI Test Maturity Matrix differently to reflect the independencies between the key areas. This is a key objective of the matrix, which is adopted in the PAM.

In addition, in some TPI key areas level A with its essential requirements is defined only at a later development stage of TPI (e.g. TPI.7 Metrics: level A at stage 5). Therefore, in some processes of the PRM additional BPs and WPs are defined at SPICE capability level 1.

The dependencies between the TPI key areas and thus the order of the improvement measures to achieve TPI level A is implemented with process priorities 1 through 4 in the PAM. Priority 1 means that the capability level 1 of this process should be sought as early as possible in the improvement project, while a process with priority 4 may be optimized later.

The other TPI levels are assigned to the SPICE PAs according to their position in the TPI Test Maturity Matrix, according to their content equivalent to the SPICE PAs, and regarding the practice of an assessment.

Figure 5 shows the assignment of TPI levels A through D to the SPICE capability levels 1 through 5 for the process TPI.7 Metrics. For the SPICE level 1 (and the PA1.1) some BPs and WPs are defined additionally (e.g. TPI.7 BP3 Use metrics in test reporting to document the measurements.)

TPI Process and Level	SPICE Process Attribute and Capability Level								
	PA1.1	PA2.1	PA2.2	PA3.1	PA3.2	PA4.1	PA4.2	PA5.1	PA5.2
	1	2		3		4		5	
7.**A** Project metrics (product)			x						
7.**B** Project metrics (process)					x				
7.**C** System metrics							x		
7.**D** Organization metrics (>1system)									x

Figure 5: Assignment of TPI levels to the SPICE capability levels (example for TPI.7 Metrics)

Process-specific BPs and WPs exist in SPICE only at capability level 1. At the higher levels there are defined GPs and GRs (SPICE-specific and not derived from TPI), which have to be implemented specific to the process. Therefore, TPI checkpoints of the higher levels are assigned to the corresponding GPs of SPICE.

3.4. The Presentation of the Assessment Result

The primary result of an assessment with a SPICE-compatible model is the process profile exemplified in Figure 6.

With the possibilities of a modern spreadsheet the TPI Test Maturity Matrix can be created automatically and parallel to the recorded estimation results for practices (BPs, GPs), WPs, and GRs and PAs (Figure 7).

Below are the equivalents of the ratings in TPI and SPICE:

TPI achieved: SPICE fully achieved (> 75%),
TPI not achieved: SPICE not, partially or largely achieved (<76%).

With the combined and synchronous presentation of the SPICE Process Profile and the TPI Test Maturity Matrix, the two main recipients of the assessment results will be addressed:

Assessment Sponsor:
Comparable presentation of the achieved capability levels of the processes through the SPICE process profile (Focus: Management attention)

Improvement Project:
Presentation of the optimization potentials in the key areas of the TPI Test Maturity Matrix (Focus: Order of measures regarding prioritization and dependencies)

The different assessed maturity levels in Figures 6 (SPICE Process Profile) and 7 (TPI Test Maturity Matrix) result from the following facts:

Additional BPs and WPs are defined in some processes of the PRM. They are corresponding to the PA 1.1 Process Performance on the SPICE capability level 1.

The graduated rating scale in SPICE leads to a more realistic and more accurate picture of the existing capability level of the process than the one in TPI with a two-step rating.

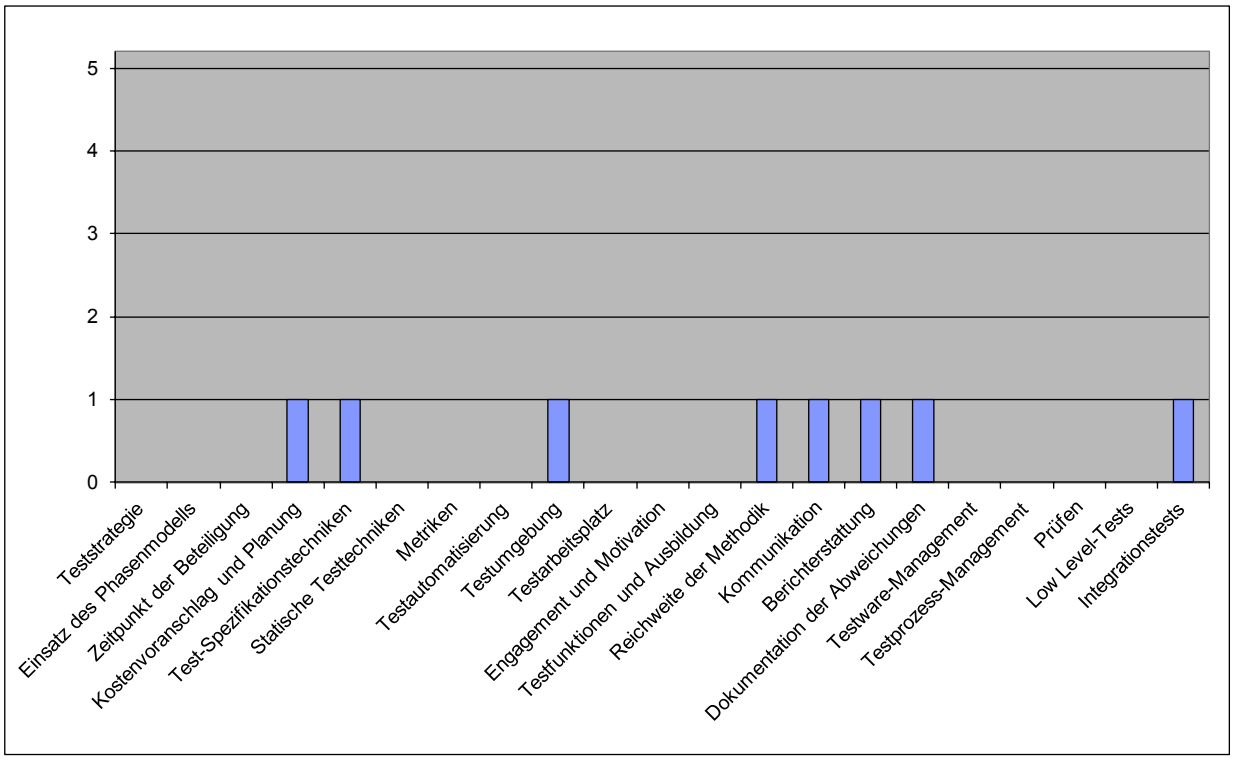


Figure 6: Example of a SPICE process profile (synchronous with figure 7)

	Key Area/Scale	0	1	2	3	4	5	6	7	8	9	10	11	12	13
			Controlled					Efficient					Optimizing		
1	Test strategy		A					B				C		D	
2	Life-cycle model		A			B									
3	Moment of involvement			A				B				C		D	
4	Estimating and planning				A							B			
5	Test specification techniques		A		B										
6	Static test techniques					A		B							
7	Metrics						A			B			C		D
8	Test automation					A			B			C			
9	Test environment				A				B						C
10	Office environment				A										
11	Commitment and motivation		A				B					C			
12	Test functions and training				A			B				C			
13	Scope of methodology					A						B			C
14	Communication			A		B							C		
15	Reporting		A			B		C					D		
16	Defect management		A				B		C						
17	Testware management			A			B				C				D
18	Test process management		A		B								C		
19	Evaluation							A			B				
20	Low-level testing					A		B		C					
21	Integration testing					A			B			C			
			Maturity Level					Improvement Step							

Figure 7: Example of a TPI Test Maturity Matrix (synchronous with Figure 6)

3.5. The Benefits of the Process Assessment Model

The following basic advantages of using a SPICE-compatible PAM for TPI in practice can be identified:

The capability level is determined in accordance with a standardized, proven, and transparent model.

The uniform number of capability levels across all processes significantly facilitates the understanding of the assessment results.

The developed PAM focuses on the capability levels 1-3, which corresponds to the existing up to the target maturity of the test process in many companies.

The use of a graduated rating scale leads to a meaningful picture of the existing maturity of the testing process.

Transferring all elements of the TPI model (in particular suggestions for improvement) into the PAM offers the security of tried and tested methods.

The use of the PAM in assessments up to capability level 3 does not lead to a significant increase in the necessary effort.

The PAM contains several additions and enhancements in comparison to the TPI model, facilitating the use in assessments.

4. The implementation of the Process Assessment Model

The PAM is implemented in an Excel workbook and therefore allows extensive design options to support the work of the assessor in carrying out the assessment and collecting and analyzing the results.

The following functions can be executed within the workbook:

- Collection of formal and organizational data for the assessment
- Definition of the scope of the assessment
- Recording of all evidences provided and evaluated
- Capture of the strengths and weaknesses of all the practices, work products, and resources
- Evaluation of all practices, work products, and resources
- Automatic calculation of achieved capability levels of all process attributes across all processes in the scope of the assessment with the possibility of manual correction
- Preparation of the corresponding SPICE Process Profile
- Preparation of the corresponding TPI Test Maturity Matrix

5. References

[1] Definition und Anwendung eines Prozess-Assessment-Modells zur Testprozessverbesserung, Sabine Link, Fernuniversität in Hagen

[2] TPI Automotive – Test Process Improvement, Sogeti Deutschland GmbH

> biography

Sven Koos
has been with bbv Software Services AG since 2009 and works as Test Manager, Quality Manager and Assessor in various projects.
He was born in Wolgast in 1966 and holds a Master in Engineering from the Technical University of Breslau, Poland. As a designated engineer in telecommunication systems, he developed for example automatic communication systems for analog and digital networks for the Deutsche Bundespost.
He was a major contributor for the planning and development of a database based control system of the sorting system of the largest European mail distribution center at Frankfurt airport. He also worked as sub project manager for data communication systems for Alcatel Transportation Systems, for instance for the development of the electronic interlocking and communication system of the ICE high speed railway Cologne – Frankfurt. His contributions were in the area of the automated monitoring of major parts of the system.
In 2005 he started his career as Software Quality Engineer and Assessor at Siemens VDO, later merged with Continental, in the area of safety-critical components of the automotive industry, specifically of the BMW Group. Since 2009 he has mainly worked as Quality and Test Manager at Swiss Federal Railways (German: SBB, Schweizerische Bundesbahnen).
Sven Koos holds also an MBA. Besides that he is an ISO 15504 (SPICE) and automotive SPICE Assessor, a SAQ/ISTQB® Certified Tester, IREB Certified Professional for Requirements Engineering, and Certified Scrum Master (CSM) and Product Owner (CPO).

Masthead

EDITOR
Díaz & Hilterscheid
Unternehmensberatung GmbH
Kurfürstendamm 179
10707 Berlin, Germany

Phone: +49 (0)30 74 76 28-0 Fax: +49 (0)30 74 76 28-99 E-Mail: info@diazhilterscheid.de

Díaz & Hilterscheid is a member of "Verband der Zeitschriftenverleger Berlin-Brandenburg e.V."

EDITORIAL
José Díaz

LAYOUT & DESIGN
Katrin Schülke

WEBSITE
www.testingexperience.com

ARTICLES & AUTHORS
editorial@testingexperience.com

450.000 readers

ADVERTISEMENTS
sales@testingexperience.com

SUBSCRIBE
www.testingexperience.com/subscribe.php

PRICE
online version: free of charge -> www.testingexperience.com
print version: 8,00 € (plus shipping) -> www.testingexperience-shop.com

ISSN 1866-5705

Picture Credits

Index of Advertisers

Madrid

JUNE 4th-7th

expoQA'12

The best International Conference on Software Testing ...under the sun!

40 International Speakers,
4 Tutorials, 4 Keynotes, 4 Tracks
Webinars, 1000m² Exhibition,
Debates, Networking, Gala Dinner

expoQA'12

Register with a special 20% discount AGREC20 at www.expoqa.com/en

A unique opportunity to share your experience with Test Experts from all around the world.

Sponsors

Endorsed by

Training with a View

Díaz Hilterscheid

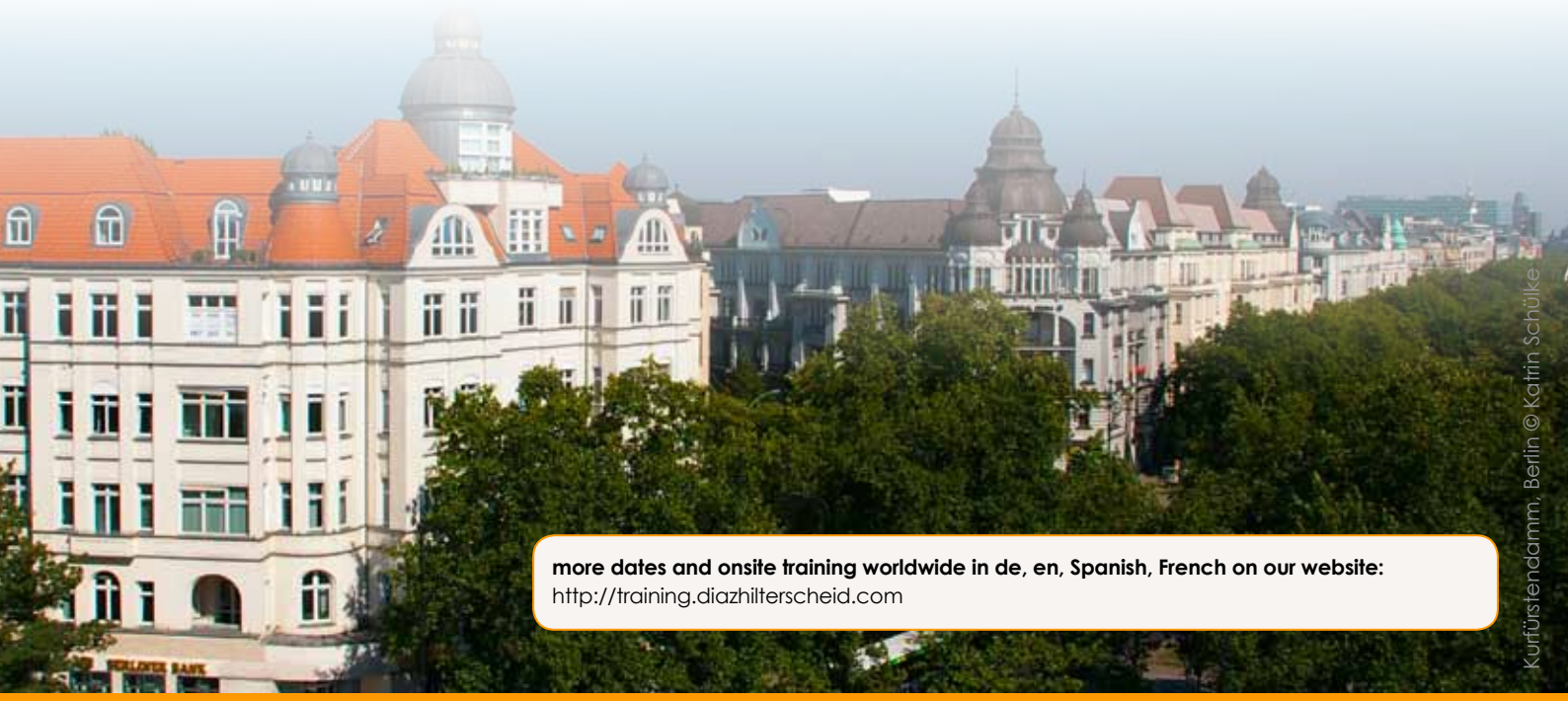
Dates *	Course	Language	Place	Price in €
22.06.12–22.06.12	Anforderungsmanagement	de	Berlin	499
24.08.12–24.08.12	Anforderungsmanagement	de	Berlin	499
23.07.12–27.07.12	CAT – Certified Agile Tester (DE)	de	München/Karlsruhe	2,100
27.08.12–31.08.12	CAT – Certified Agile Tester (DE)	de	Frankfurt	2,100
11.06.12–15.06.12	CAT – Certified Agile Tester (EN)	en	Stockholm, Sweden	2,100
02.07.12–06.07.12	CAT – Certified Agile Tester (EN)	en	Orlando, USA	2,100
20.08.12–21.08.12	HP Quality Center	de	Berlin	998
06.06.12–08.06.12	IREB® Certified Professional for Requirements Engineering – Foundation Level (DE)	de	Berlin	1,400
09.07.12–11.07.12	IREB® Certified Professional for Requirements Engineering – Foundation Level (DE)	de	Mödling, Austria	1,400
19.06.12–21.06.12	IREB® Certified Professional for Requirements Engineering – Foundation Level (EN)	en	Oslo, Norway	1,600
16.07.12–18.07.12	ISTQB® Certified Tester Foundation Level – Kompaktkurs (DE)	de	München/Karlsruhe	1,495
23.07.12–25.07.12	ISTQB® Certified Tester Foundation Level – Kompaktkurs (DE)	de	Berlin	1,495
06.08.12–08.08.12	ISTQB® Certified Tester Foundation Level – Kompaktkurs (DE)	de	Hamburg	1,495
18.06.12–21.06.12	ISTQB® Certified Tester Foundation Level	de	Berlin	1,800
18.06.12–21.06.12	ISTQB® Certified Tester Foundation Level	de	Mödling, Austria	1,800
02.07.12–05.07.12	ISTQB® Certified Tester Foundation Level	de	Leipzig	1,800
30.07.12–02.08.12	ISTQB® Certified Tester Foundation Level	de	Frankfurt	1,800
13.08.12–16.08.12	ISTQB® Certified Tester Foundation Level	de	Berlin	1,800
27.08.12–30.08.12	ISTQB® Certified Tester Foundation Level	de	Köln/Düsseldorf	1,800
04.06.12–08.06.12	ISTQB® Certified Tester Advanced Level – Technical Test Analyst	de	Berlin	2,100
09.07.12–13.07.12	ISTQB® Certified Tester Advanced Level – Technical Test Analyst	de	Frankfurt/Stuttgart	2,100
30.07.12–03.08.12	ISTQB® Certified Tester Advanced Level – Test Analyst	de	Berlin	2,100
18.06.12–22.06.12	ISTQB® Certified Tester Advanced Level – Test Manager	de	München	2,100
25.06.12–29.06.12	ISTQB® Certified Tester Advanced Level – Test Manager	de	Berlin	2,100
20.08.12–24.08.12	ISTQB® Certified Tester Advanced Level – Test Manager	de	Stuttgart/Karlsruhe	2,100

IREB International Requirements Engineering Board

CAT Certified Agile Tester®

ISTQB® Accredited Training Provider

*** subject to modifications**
all prices plus VAT



more dates and onsite training worldwide in de, en, Spanish, French on our website:
http://training.diazhilterscheid.com

Kurfürstendamm, Berlin © Katrin Schülke