OULUN SEUDUN
AMMATTIKORKEAKOULU

Virpi Mehtälä

AGILE TESTING – HOW A TESTER'S ROLE IS CHANGING IN SOFTWARE DEVELOPMENT PROJECTS

AGILE TESTING – HOW A TESTER'S ROLE IS CHANGING IN SOFTWARE DEVELOPMENT PROJECTS

Virpi Mehtälä
Master's Thesis
2012
Degree Programme in Industrial
Management
Oulu University of Applied Sciences

ABSTRACT

Oulu University of Applied Sciences
Degree programme in Industrial Management, Master of Engineering

Author(s): Virpi Mehtälä
Title of thesis: Agile Testing – How a Tester's Role Is Changing in Software Development Projects
Supervisor(s): Kari Laitinen, Esa Nakkila
Term and year when the thesis was submitted: Spring 2012 Number of pages: 45

ABSTRACT

Software production has changed in recent years to meet the need of getting the products faster to markets. A big challenge is to find the most suitable people for the agile development work.

To change the way of doing projects, the agile does not happen only by request. On the other hand, markets require fast and flexible changes. This dilemma raised a question of what kind of people can survive best in this change. As the Agile methods were taken into use in software development, a fear was emerged that testing work is too expensive to survive in western countries.

The objective of this thesis work was to study literature to find out what kind of people are needed as software testers in the agile development framework and how these people can be recruited.

Agile software development includes testing as a part of the development, not as a separate phase of the developing work. Testing is not necessarily visible in the organizational chart, but the actual work is still there, and its importance for the development work is bigger than ever. Moreover, the same reason why there was a need for dedicated testers earlier is still valid.

There is still an open question how to train these valuable testers to meet the present and forthcoming requests. The school system does not provide much, and training inside the companies is limited.

Keywords: *Software development, testing, agile, SCRUM, KANBAN, quality control*

PREFACE

The topic for this master’s thesis work was discussed with Hanna Hyry, site manager at Ixonos, in 2010. The preconception was then that the ICT industry will change and that the change will require certain changes for the work and the working people. I have pursued the ICT industry change in my work within two projects since 2010 and studied the literature aside. The pace of that change became faster during 2011 and I started to see what kind of people the new industry may need. I wrote this thesis mostly in Berlin, and I would like to thank the people who helped me to clarify my thoughts by long discussions. Many thanks to my collegues in Berlin, Jussi Silfer, Budi Budiardjo, Jaroslaw Wyrzytowicz and Michal Marzec. Special thanks go to my pre-reader Esa Nakkila and of course to my family for their understanding and support along the way.

Oulu, Finland, February 2012

Virpi Mehtälä

TABLE OF CONTENTS

Used Acronyms

ANSI	American National Standards Institute
AST	The Association for Software Testing
BCS	British Computer Society
CAT	Certified Agile Tester
CMM	Capability Maturity Model
CMMI	Capability Maturity Model Integration
IEEE	Institute of Electrical and Electronics Engineers
ISEB	The Information Systems Examinations Board
ISO	International Organization for Standardization
ISTQB	The International Software Testing Qualifications Board
iSQI	The International Software Quality Institute
KANBAN	Form of agile project management
SCRUM	Form of agile project management
TMM	Test Maturity Model

1 INTRODUCTION

Software production has turned to agile development to deliver small iterations of the business value in extremely short cycles. People who produce software have been learning new habits of doing their everyday work. Learning causes pain when there are teams and whole organizations who have to learn at the same time. While learning new ways to do the production, there is a question: what to do with all the old ways? There are new roles, and some old roles do not exist anymore. For example, testers have become team members. Are testers obsolete in the agile world? What kind of people can manage these new roles best?

The objective of this thesis work was to study literature to find out what kind of people are needed as software testers in the agile development framework and how these people can be recruited.

This thesis work focuses on the software tester’s qualifications, knowledge, and skills. It concentrates on the knowledge that turns to skills and abilities. The tester’s personality is on the focus only on attitude; other psychological aspects are excluded from this thesis work.

This thesis is qualitative in nature with some inductive reasoning, descriptive and normative by given recommendations. This is a pure theoretical research based on literature, and steered with the author’s own empirical experiences during two multi-team projects.

2 SOFTWARE DEVELOPMENT

Software development is systematic work. The more a complex system is under development, the easier it runs into a chaos. To avoid that chaos to happen, the software production people have learnt their lessons and designed software development models to help. Each new model should be better than the older one and each of them have some kind of a process to follow.

2.1 Traditional Software Development

The traditional Waterfall model, illustrated in Figure 1, is a typical linear software life cycle model, where one stage concludes when the other starts. If the previous stage is not completed, the next stage cannot start. The testing starts when the implementation is done in the Verification stage. In the Waterfall-model, the testing happens late in the software life cycle and the errors are difficult and expensive to fix.

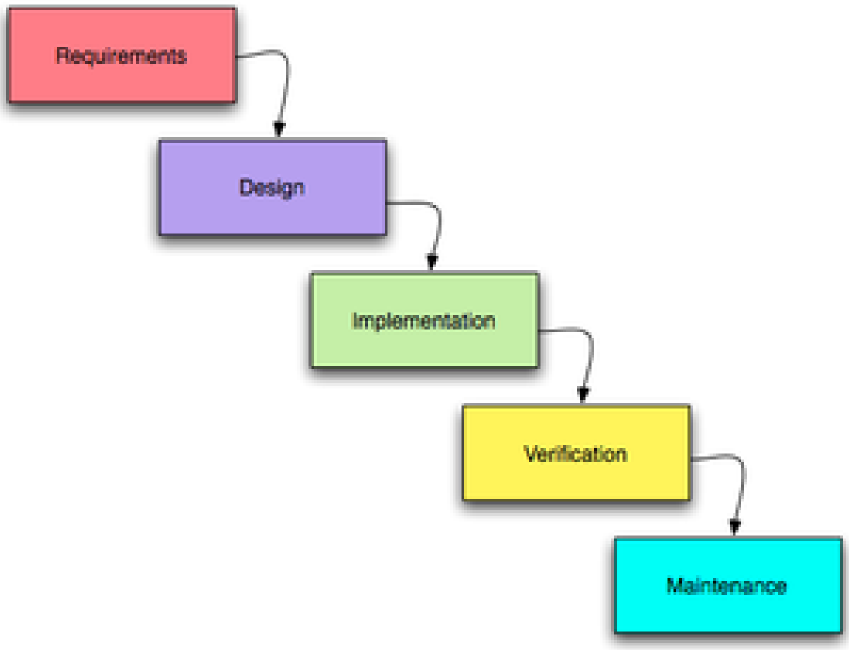


FIGURE 1. Waterfall model
(Waterfall model from Wikipedia 19 January 2012.)

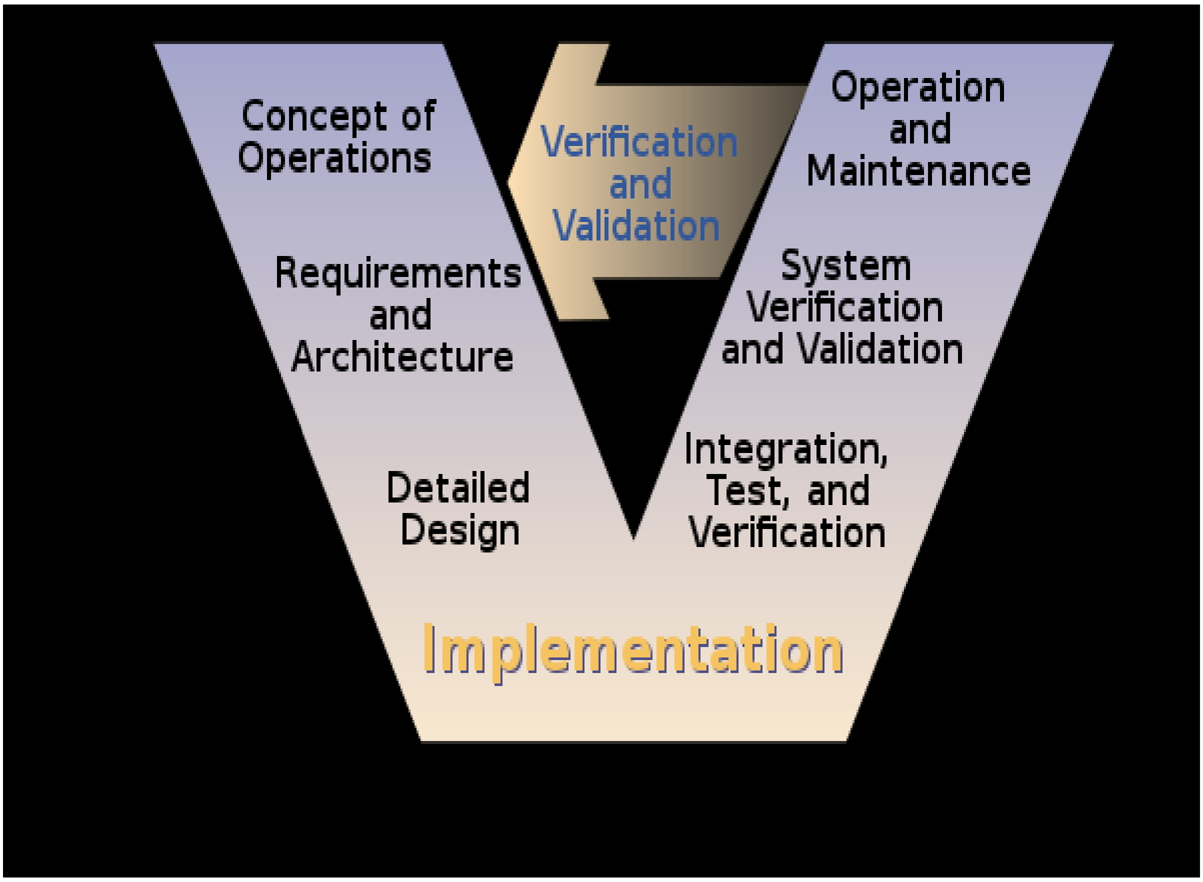

Concept of Operations
Requirements and Architecture
Detailed Design
Verification and Validation
Operation and Maintenance
System Verification and Validation
Integration, Test, and Verification
Implementation

FIGURE 2. V- model
(V-Model from Wikipedia, 19 January 2012.)

V-model, illustrated in Figure 2, was developed to bring the testing closer to each development phase. V-model defines common practices and roles in a project. A new problem in V-model compared to Waterfall is that V-model also requires the stakeholders' presence and continuous communication during the project.

V-model has corresponding test plans for each development phase. The verification means that the test results of each phase compare to the corresponding development phase documents. Validation means that system and acceptance testing compare the software to the user's requirements.

V-model is flexible to different project types, and the sizes and costs in each phase are easy to predict with this model.

2.2 Agile Manifesto

By the time, software development became more and more complex and time consuming, as V-model was too slow to produce software as the business wanted. Therefore something more efficient and faster was required, and that was how the Agile Manifesto was born.

“The Agile Manifesto was written in February of 2001, at a summit of seventeen independent-minded practitioners of several programming methodologies. The participants didn't agree about much, but they found consensus around four main values.” (Beck et al., 19. January 2012)

The main idea for Agile Manifesto was to find the most time-consuming non-valuable parts in the model, and prevent them to happen.

> We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:
>
> - Individuals and interactions over processes and tools
> - Working software over comprehensive documentation
> - Customer collaboration over contract negotiation
> - Responding to change over following a plan
>
> That is, while there is value in the items on the right, we value the items on the left more. (Beck et al., 19. January 2012)

The common Principles of Agile Manifesto are the following:

- The highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- Changing requirements are always welcome, even late in the development.
- Deliver only working software frequently, with a preference to the shorter timescale.
- Business people and developers must work together daily throughout the project.
- Build projects around motivated and skilful individuals.
  Give them the environment and support they need, and trust them to get the job done.
- Emphasize face-to-face conversation.
- Working software is the primary measure of progress.
- Agile processes promote sustainable development.
- Continuous attention to technical excellence and good design enhance agility.

- The best architectures, requirements, and designs emerge from self-organizing teams
- Continuous improvement

(Beck et al., 19. January 2012

These principles are demanding for the people who are working in an agile team. For a successful project, the team members have to have discipline to follow these principles and knowledge of suitable methods and techniques to achieve it.

Agile Framework does not give instructions of what to do in a project, but it gives principles how to do the needed actions to gain the target. Each project has to decide what these needed actions are as well as which agile process they want to use. In Finland, the most used agile processes are SCRUM and Kanban.

2.2.1 SCRUM

Scrum is a framework within which the software development work happens. It is not a methodology, and it requires that the Scrum-team continuously improves itself. Developing happens in iterations.

“Scrum does not prescribe. Scrum includes general guidelines about how to do development and principles to be applied when these recommendation are insufficient. This means that people need to learn think differently.” (Schwaber, K. 2007, xi)

In Figure 3 are illustrated the standard Scum process, its artefacts, roles and, processes. Scrum roles are four in this figure, but the pure Scrum has only three roles: Product Owner, Scrum Master and Team Member. Team members have different skills, for example QA and developer.

Every Scrum team, regardless of its level in the enterprise, will go through the steps of forming, storming, norming and performing. (Schwaber, K. 2007, 75) Basic knowledge of team developing steps is essential to every team member in an agile developing team.

It is also important to remember that steps are not straightforward, whatever changes in the team; it changes easily the step back or forth. Scrum Master does not need so much effort to solve these problems if the team members are aware of these steps.

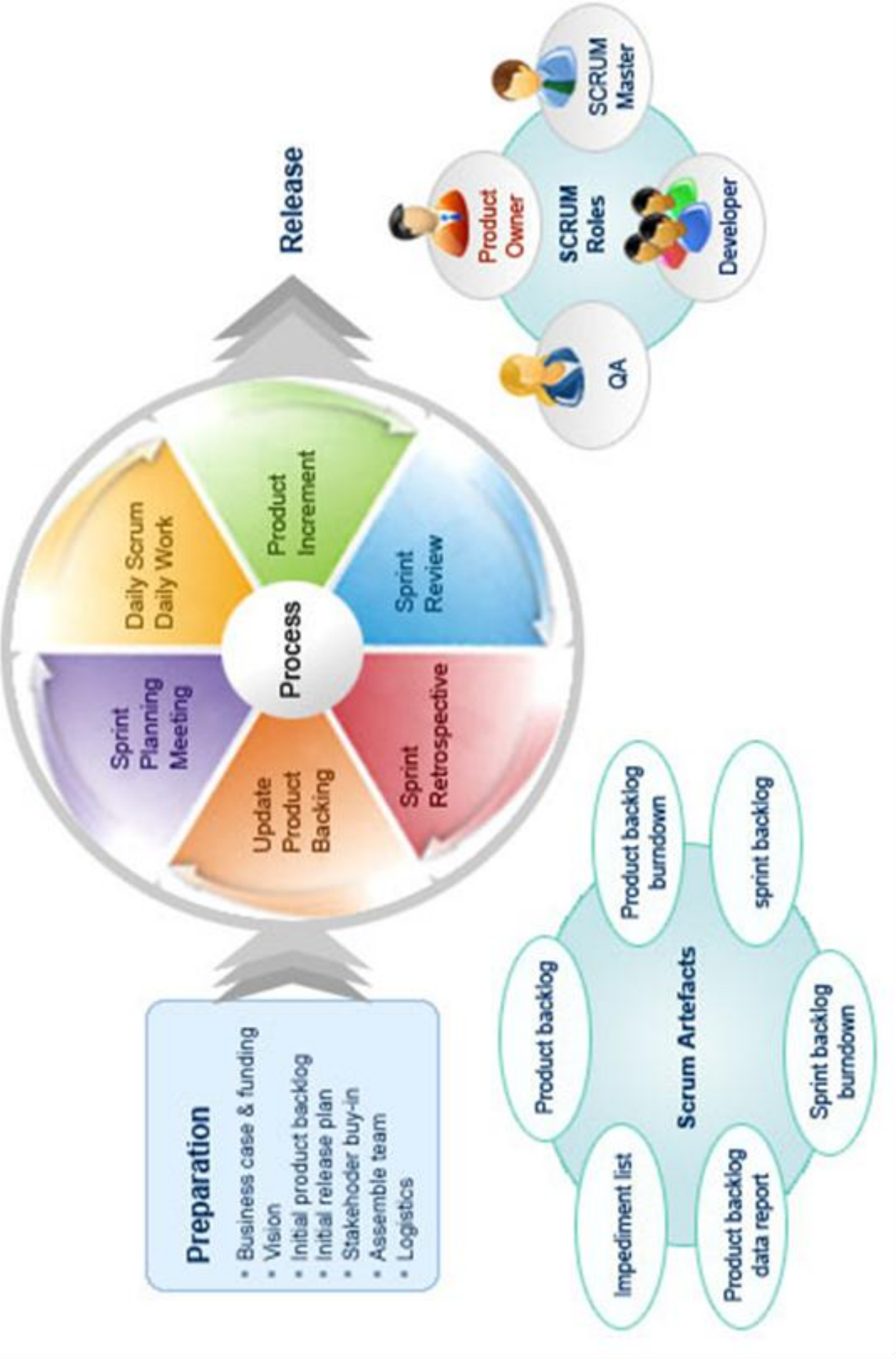
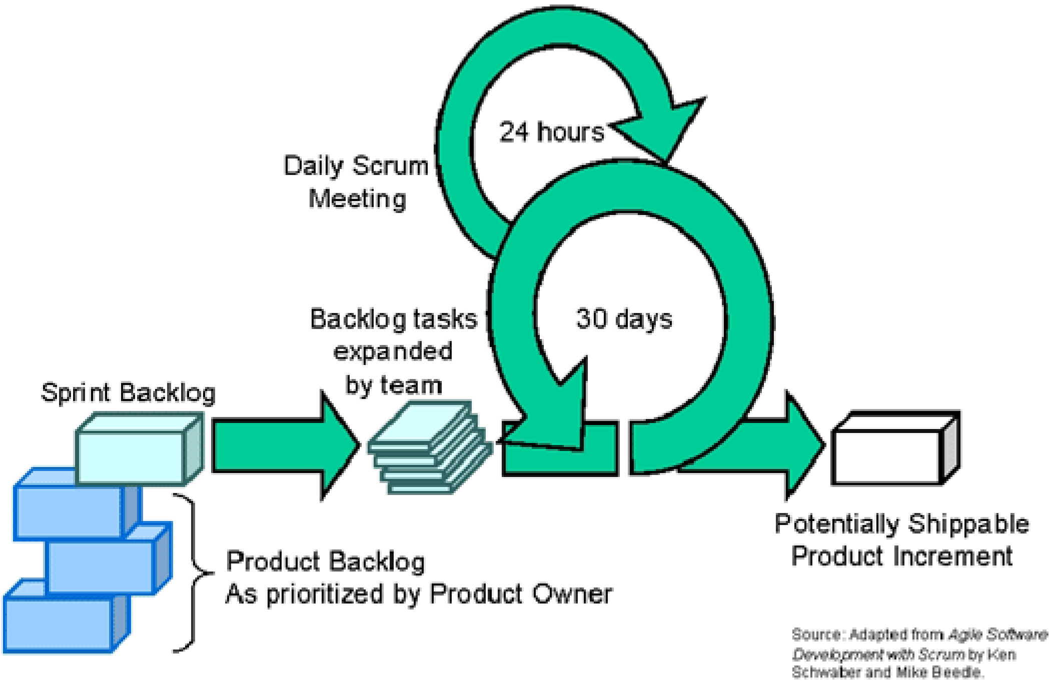


FIGURE 3. Standard Scrum process

(PPM Studio::Agile Project Management Software for Agile software development , 19. January 2012.)

Scrum has only few primary practices, illustrated in Figure 4, and coding or testing are not among them as they are in Waterfall or V-model.


24 hours
Daily Scrum Meeting
30 days
Backlog tasks expanded by team
Sprint Backlog
Product Backlog
As prioritized by Product Owner
Potentially Shippable Product Increment

Source: Adapted from *Agile Software Development with Scrum* by Ken Schwaber and Mike Beedle.

FIGURE 4. Scrum primary practices

In Scrum designing, implementing and verifying tasks are continous, not a specific phase of developing. Developing happens in specified Sprints, and each Sprint's outcome is a potentially shippaple delivery. In Figure 4, the Sprint length is 30 days, but can be whatever the team decides. The recommendation for a Sprint length is from two to four weeks. The project continues developing Sprint by Sprint as long as the Product Backlog is empty.

Daily Scrum Meeting is for team members to tell each other what they have done since the latest Daily and what they will do till the next Daily and if they have any impediments which prevent them to do their work. Team members with testing skills have their testing tasks in Backlog as the other team members and testers plan, execute and report their tasks according to the team manners.

2.2.2 Kanban

Kanban is less structured than Scrum. It is no process framework at all, but a model for introducing a change through incremental improvements. Kanban emphasizes WIP, Work In Progress. Kanban is Japanese and means a Visible Card, which states the tasks in the Product Backlog and will be on the Board when decided to start implementing them. Figure 5 shows the simplest Kanban Board. (Kniberg, H. & Skarin, M. 2010, 46.)

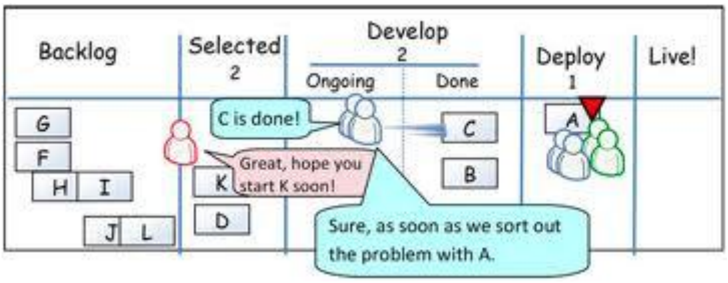

Backlog
Selected
2
Develop
2
Ongoing
Done
Deploy
1
Live!
G
F
H
I
J
L
K
D
C
B
A
C is done!
Great, hope you start K soon!
Sure, as soon as we sort out the problem with A.

FIGURE 5. Kanban Board

A project can have as many Kanban Boards as needed, and more is not better than a few, but each project can decide what they need. The Board makes the workflow visible and the project steering easier. Kanban seems simple as it has only few principles, but it requires more discipline to success than Scrum.

Kanban principles:

- Visualize the workflow
- Limit work-in-progress
- Manage flow
- Make Process Policies Explicit
- Improve Collaboratively

Testing can be part of Decision of Done and included in the Work In Progress- state with the implementation work. In addition, testing can have its own Kanban Board that has only testing tasks. Kniberg and Skarin have collected the main differences in table 1.

TABLE 1. Differences between Scrum and Kanban (Kniberg, H. & Skarin, M. 2010, 50)

Scrum	**Kanban**
Timeboxed iterations prescribed.	**Timeboxed iterations optional.** Can have separate cadences for planning, release, and process improvement. Can be eventdriven instead of timeboxed.
Team commits to a specific amount of work for this iteration.	**Commitment optional.**
Uses **Velocity** as default metric for planning and process improvement.	Uses **Lead time** as default metric for planning and process improvement.
Cross-functional teams prescribed.	Cross-functional teams optional. **Specialist teams allowed.**
Items must be broken down so they can be completed within 1 sprint.	No particular item size is prescribed.
Burndown chart prescribed	No particular type of diagram is prescribed
WIP limited indirectly (per sprint)	**WIP limited directly** (per workflow state)
Estimation prescribed	**Estimation optional**
Cannot add items to ongoing iteration.	**Can add new items whenever capacity is available**
A **sprint backlog is owned by one specific team**	A **kanban board may be shared by multiple teams** or individuals
Prescribes 3 roles (PO/SM/Team)	**Doesn't prescribe any roles**
A **Scrum board is reset** between each sprint	A **kanban board is persistent**
Prescribes a prioritized product backlog	**Prioritization is optional**

3 WHAT IS TESTING?

Definition of testing has varied over the years. Glenford J. Myers initially introduced the separation of debugging from testing in 1979 in his book *The Art of Software Testing* as follows: “Testing is the process of executing a program or system with the intent of finding errors” (Myers 1979, 6.). As the statement says, the main purpose of testing was finding errors at the end of development work.

In 1983, Bill Hezel included quality assessment and defined testing in his book *The Complete Guide to Software Testing* as follows: “Testing is any activity aimed at evaluating an attribute of a program or system. Testing is the measurement of software quality”. (Hezel 1988, 6, 242.) Testing was as a tool for quality control, and the quality of the delivered software was the test organization’s main responsibility. This approach became inefficient and expensive, and as developers started to use some tools and techniques to keep the discipline in code quality, testing was an inevitable expense in software development business. By the time the significance of testing has increased due to shortened design lifecycles, decreased time-to market and more complex systems (Belt 2009, 13-14).

Appreciation of testing has been low; thus, no software producer dare deliver software without testing. Destructive approach to software is inconvenient.

Testing is the only role on the project that does not directly focus on success. Everyone else creates something or creatively guides its creations. However, testers are *negative.* This can be a depressing job, almost like a parody of a Greek myth:”On the island of the testers, they were doomed forever to search for what could not and should not exist, knowing that to succeed would bring misery to the Gods.” (Bach et al 2002, 151.)
The definition of testing has become more open to consider the multiple dimensions of the testing. Today it is defined as follows: “We believe great software testing requires Craftsmanship, Science, and Passion” (Association for Software Testing, 26. November 2011)

3.1 Testing Cycle

There is a typical cycle for software testing. This sample is useful regardless of the used development model. The Waterfall model has particular testing phase, the V-model has grouped the phases according to its development model. In the Agile testing the testers use these same phases to conduct the testing even if they emphasise certain phases more than others or some remain invisible.

- **Requirements analysis**: During the design phase, testers work with developers and designers to determine what aspects of a design are testable and with what parameters those tests work.
- **Test planning**: Test strategy, test plan, test ware creation.
- **Test development**: Test procedures, test scenarios, test cases, test datasets, test scripts to use in testing software.
- **Test execution**: Testers execute the software based on the plans and test documents.
- **Test reporting**: Testers generate metrics and make final reports
- **Test result analysis**: Or Defect Analysis, decisions to actions
- **Regression testing**: Testers execute this in order to ensure that the software product as a whole is still working correctly.
- **Test Closure**: Once the test meets the exit criteria.

These phases recur in every planned testing occasion. Such occasion may happen only on the testers mind, and professional tester recognizes these phases.

3.2 Associated with Testing

Testing activity is not self-sufficient, but it is done for request, and the outcome is valuable for many different groups of operators and operations. Many factors have their direct or indirect impact on testing and testers.

Regardless of the purpose of the developed software, the outcome has a recipient who is interested in the deliverable. The customer is interested in testing as it reveals the status of the product, and the software user is interested when the product is in production. Management for its part is interested as for the fulfillment of the organization’s mission. (Perry, W. 2000, 34-35.)

Particularly a commercial software customer awaits the profit, but a proprietary software customer is also keen on the costs. The business gives its own perspective for testing by business risk mitigation. The risk is a probability that undesirable events become true. Controls are the means used by organizations to minimize the risk. Software testing is a control. (Perry, W. 2000, 8.)

Testing is related to software development. Testers and developers have the same target. The factors that direct programming work have also influence on testing work. The most important factors about a software development process are (1) there is a process, (2) it is understood, (3) it is followed. (Beizer, B. 1995, 13.) Less important is which the chosen process model is, as the process model itself does not lead to success or failure. Cultural, ethic, application, and national specifics have a greater influence on the process than grandiose process theories. (Beizer, B. 1995, 14.)

3.3 Tester as a Title and a Career

Until the 1980s the term "*software tester*" was used generally, but later it was also seen as a separate profession. Dave Gelperin and William C. Hetzel classified in *The Growth of Software Testing* 1988 the phases and goals in software testing in the following stages:

Until 1956 - Debugging oriented
1957–1978 - Demonstration oriented
1979–1982 - Destruction oriented
1983–1987 - Evaluation oriented
1988–2000 - Prevention oriented

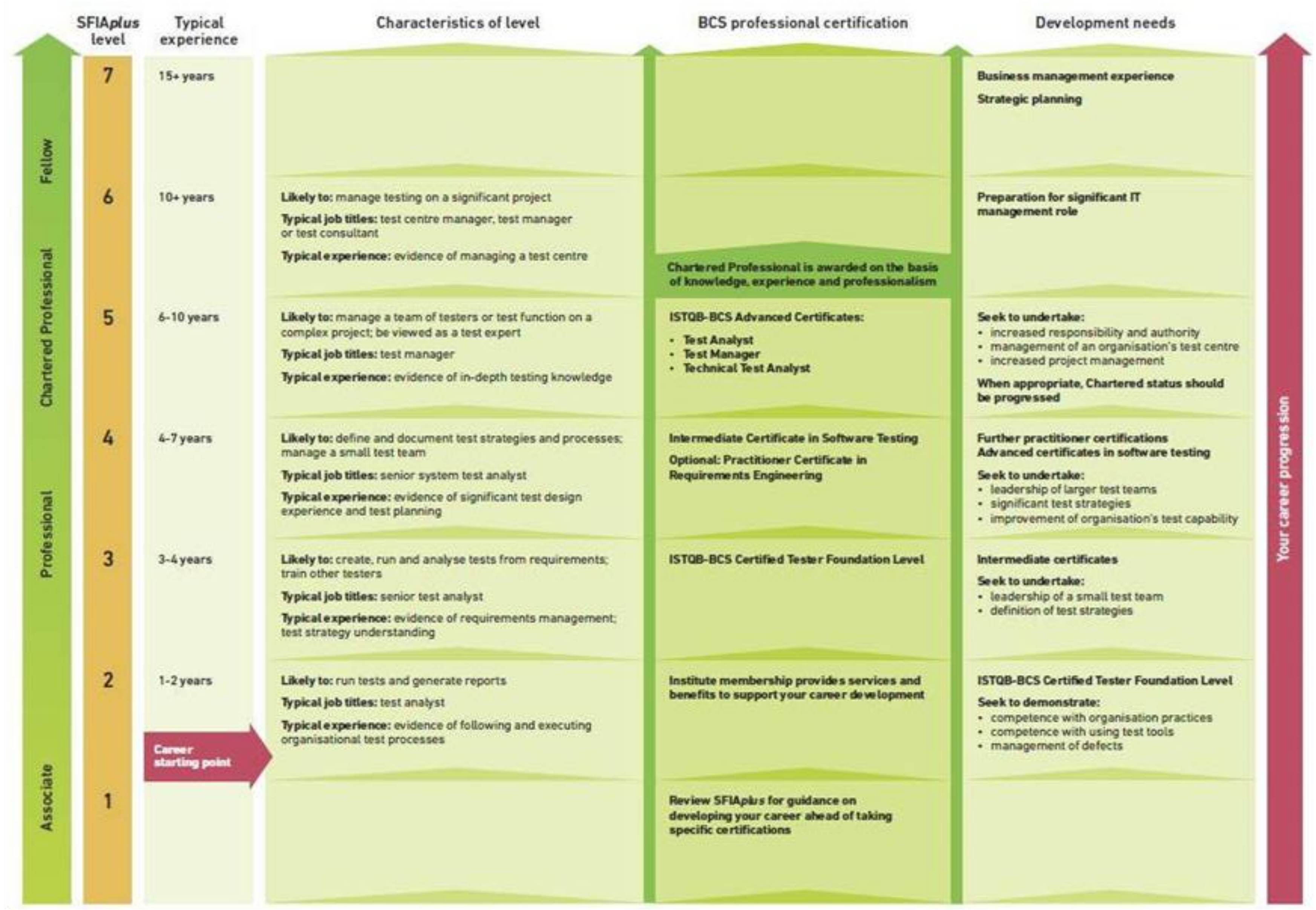
Regarding the periods and the different goals in software testing, different roles have been established: *manager, test lead,* test *designer, tester*, *automation developer,* and *test administrator*.

In the 2000s-the following incidents have had their consequences to the tester's roles and statuses:

2001 Agile Manifesto is published

2002 ISTQB is founded
(Meerts, 26. November 2011)

Where the Agile software development framework aligns testers as team members with special testing skills, the ISEB/ISTQB certification training materials divide testers to different defined categories and career paths.



	SFIA*plus* level	Typical experience	Characteristics of level	BCS professional certification	Development needs
Fellow	7	15+ years			**Business management experience** **Strategic planning**
Chartered Professional	6	10+ years	**Likely to:** manage testing on a significant project **Typical job titles:** test centre manager, test manager or test consultant **Typical experience:** evidence of managing a test centre	**Chartered Professional is awarded on the basis of knowledge, experience and professionalism**	**Preparation for significant IT management role**
	5	6-10 years	**Likely to:** manage a team of testers or test function on a complex project; be viewed as a test expert **Typical job titles:** test manager **Typical experience:** evidence of in-depth testing knowledge	**ISTQB-BCS Advanced Certificates:** • **Test Analyst** • **Test Manager** • **Technical Test Analyst**	**Seek to undertake:** • increased responsibility and authority • management of an organisation's test centre • increased project management **When appropriate, Chartered status should be progressed**
Professional	4	4-7 years	**Likely to:** define and document test strategies and processes; manage a small test team **Typical job titles:** senior system test analyst **Typical experience:** evidence of significant test design experience and test planning	**Intermediate Certificate in Software Testing** **Optional: Practitioner Certificate in Requirements Engineering**	**Further practitioner certifications** **Advanced certificates in software testing** **Seek to undertake:** • leadership of larger test teams • significant test strategies • improvement of organisation's test capability
	3	3-4 years	**Likely to:** create, run and analyse tests from requirements; train other testers **Typical job titles:** senior test analyst **Typical experience:** evidence of requirements management; test strategy understanding	**ISTQB-BCS Certified Tester Foundation Level**	**Intermediate certificates** **Seek to undertake:** • leadership of a small test team • definition of test strategies
Associate	2	1-2 years	**Likely to:** run tests and generate reports **Typical job titles:** test analyst **Typical experience:** evidence of following and executing organisational test processes	**Institute membership provides services and benefits to support your career development**	**ISTQB-BCS Certified Tester Foundation Level** **Seek to demonstrate:** • competence with organisation practices • competence with using test tools • management of defects
	1	**Career starting point**		**Review SFIA*plus* for guidance on developing your career ahead of taking specific certifications**	

Your career progression

FIGURE 6. Software Testing Career Path by BCS (BCS Certifications, 15. November 2011)

Figure 6 describes how ISEB- certification trainings can develop the applicant's career. It emphasizes the different responsibility roles. This career path illustrates the potential career progression one can make in software testing.

The career paths structure relies on roles so that responsibility and experience increase upwards. In an agile team, the roles are different and the position or role does not describe the work. The team describes which kind of qualities and skills it needs. This might be difficult to apply for a former manager, or a test engineer who waits for a promotion. The HR department can be either a significant help or a hindrance with these problems. Human resources problems might be encountered that involve reporting structures, periodic performance reviews, handling performance problems, and determining career paths. (Cohn, M. 2010, 406.)

3.4 Certificates

A certification identifies an individual as a quality assurance leader. A certification earns the candidate the respect of colleagues and managers. One or more of these certifications is frequently a prerequisite for promotion or acquiring a new position. The certification is a formal recognition of a level of proficiency in the information technology (IT) quality assurance industry. The recipient is acknowledged as having an overall comprehension of the disciplines and skills represented in a comprehensive body of knowledge for a respective software discipline. In Finland, ISEB / ISTQB is the most recognized software testing certificate institution. ITIL is a certification program for services.

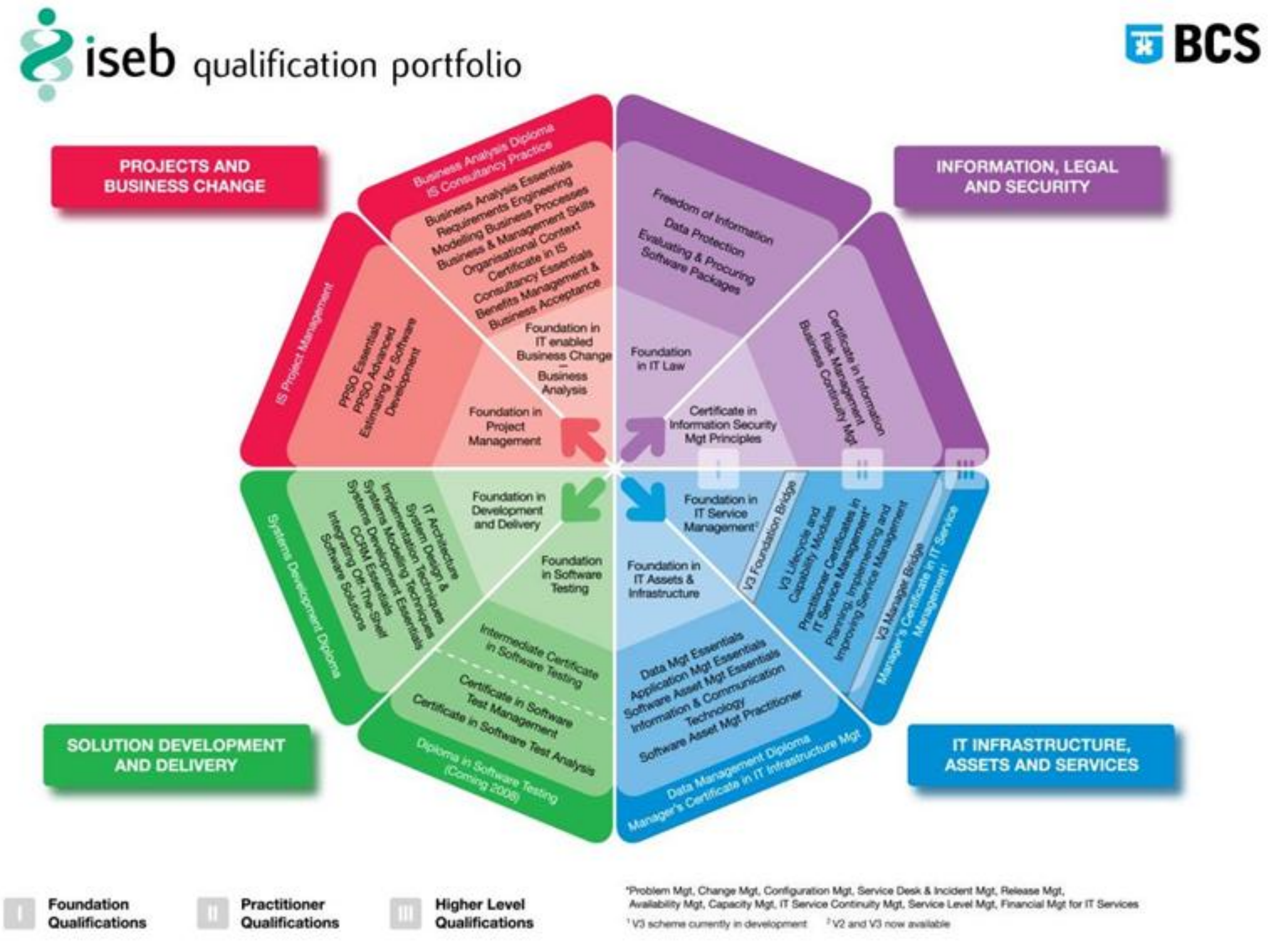
The Association for Software Testing (AST), founded in the USA in 2004, is a professional non-profit association that is dedicated to build a testing community that views the role of testing as skilled, relevant, and essential to the production of faster, better, and less expensive software products. (Association for Software Testing, 06. November 2011.) AST also provides trainings for software testers from the context driven perspective. AST has gathered together famous testing literature authors and professionals to teach and coach enthusiastic testers.

BCS, British Computer Society was founded in 1957 when The London Computer Group merged with an association of scientists to become the British Computer Society Ltd. (History of BCS| Role and Purpose, 06.November 2011.) The Institute collaborates with the government, industry and relevant bodies to promote good working practices, codes of conduct, skills frameworks and common standards. It provides practical support and information services to its members and volunteer communities, for example education, trainings and discussion groups.

The International Software Quality Institute (iSQI GmbH) provides certifications for Software Test and Quality Assurance among a wide range of certifications through professions in the software industry. (Home - ISTQB International Software Testing Qualifications Board, 06. November 2011.)

3.4.1 ISEB

The Information Systems Examinations Board (ISEB) is an examination awarding body and a part of British Computer Society. ISEB is responsible for the certificate exams; BCS is the institute to provide the trainings and grants the certificates to applicants who pass the exam. The certificate portfolio is large, as Figure 7 illustrates.


iseb qualification portfolio
BCS
PROJECTS AND BUSINESS CHANGE
INFORMATION, LEGAL AND SECURITY
SOLUTION DEVELOPMENT AND DELIVERY
IT INFRASTRUCTURE, ASSETS AND SERVICES
Business Analysis Diploma
IS Consultancy Practice
Business Analysis Essentials
Requirements Engineering
Modelling Business Processes
Business & Management Skills
Organisational Context
Certificate in IS
Consultancy Essentials
Benefits Management & Business Acceptance
Foundation in IT enabled Business Change – Business Analysis
Foundation in Project Management
IS Project Management
PPSO Essentials
PPSO Advanced
Estimating for Software Development
Freedom of Information
Data Protection
Evaluating & Procuring Software Packages
Foundation in IT Law
Certificate in Information Security Mgt Principles
Certificate in Information Risk Management
Business Continuity Mgt
I
II
III
Foundation in IT Service Management²
V3 Foundation Bridge
V3 Lifecycle and Capability Modules
Practitioner Certificates in IT Service Management*
Planning, Implementing and Improving Service Management
V3 Manager Bridge
Manager's Certificate in IT Service Management¹
Foundation in IT Assets & Infrastructure
Data Mgt Essentials
Application Mgt Essentials
Software Asset Mgt Essentials
Information & Communication Technology
Software Asset Mgt Practitioner
Data Management Diploma
Manager's Certificate in IT Infrastructure Mgt
Foundation in Development and Delivery
Foundation in Software Testing
IT Architecture
System Design & Implementation Techniques
Systems Modelling Techniques
Systems Development Essentials
CCRM Essentials
Integrating Off-The-Shelf Software Solutions
Systems Development Diploma
Intermediate Certificate in Software Testing
Certificate in Software Test Management
Certificate in Software Test Analysis
Diploma in Software Testing (Coming 2008)
I Foundation Qualifications
II Practitioner Qualifications
III Higher Level Qualifications

*Problem Mgt, Change Mgt, Configuration Mgt, Service Desk & Incident Mgt, Release Mgt, Availability Mgt, Capacity Mgt, IT Service Continuity Mgt, Service Level Mgt, Financial Mgt for IT Services

¹V3 scheme currently in development ²V2 and V3 now available

FIGURE 7. ISEB qualification portfolio by BCS
(BCS Certifications, 15. November 2011)

3.4.2 ITIL Qualification Scheme

ITIL (IT Infrastructure Library) is the collection of the best practices in the IT service management. A comprehensive qualifications scheme, accredited training organisations, and implementation and assessment tools support it. The key principles for the IT management in ITIL are as follows:

- Service Strategy
- Service Design
- Service Transition
- Service Operation
- Continuous Service Improvement

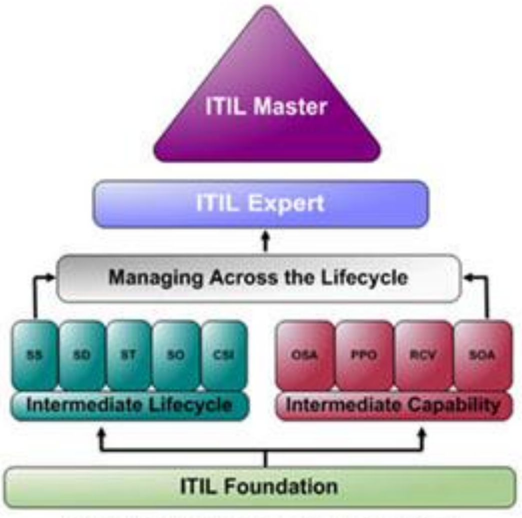
(The Official ITIL® Website, 15. November 2011)

The ITIL Qualifications scheme provides a modular approach to the ITIL framework, and is comprised of a series of qualifications focused on different aspects of ITIL Best Practice, to various degrees of depth and detail. There are four levels of qualifications within the scheme:

- ITIL Foundation
- ITIL Intermediate Level
- ITIL Expert Level
- ITIL Master Qualification

(The Official ITIL® Website, 15. November 2011)

Figure 8 illustrates the ITIL Qualifications scheme structure.



©APM Group-The Accreditor Limited 2011

FIGURE 8. ITIL qualification scheme

3.4.3 ISTQB

The other institution besides the BCS that provides training for certificates is ISTQB. “The International Software Testing Qualifications Board ® (ISTQB) is a not-for-profit association legally registered in Belgium. It is composed of 47 member Boards from worldwide countries.” (Home - ISTQB International Software Testing Qualifications Board, 06. November 2011.)

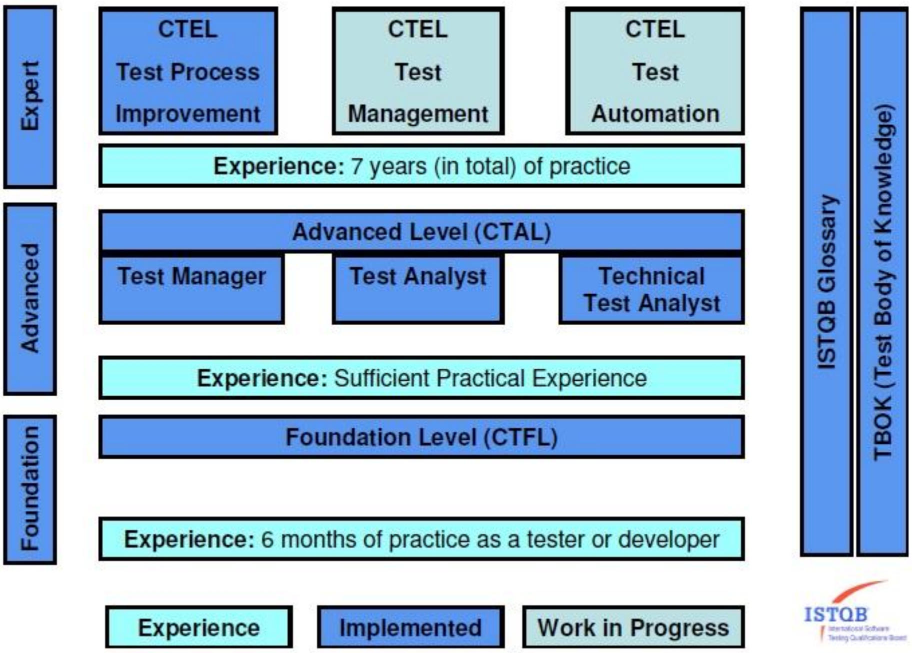


FIGURE 9. ISTQB certificates and certificate levels (Home - ISTQB International Software Testing Qualifications Board, 06. November 2011.)

The ISEB and ISTQB foundation level certificates are combining the ISEB/ISTQB certificate. ISTQB provides more choices, showed in Figure 9, to applicants to specialize than ISEB.

3.4.4 SQI CAT - Certified Agile Tester

SQI CAT Certified Agile Tester is a test certification for testers to meet the requirements of the agile development world. It provides structure, discipline, common language and, methods to work as a tester in an agile development team.

The International Software Quality Institute (iSQI GmbH) is a leading provider of certification examinations all over the world. The CAT certificate has Foundation Level certification available, and Advanced Level certification is in preparation. (Certificates – iSQI – Certifying People all over the world, 19.November 2011.)

3.5 Standards

As listed in Table 2, there are many standards related to software testing. These standards are for software development and are useful of pointing out the improvement needs for a test process. These standards are not very useful in specifying testing. Testing Maturity Modell, TMM^{SM}, introduces a set of maturity levels for software testing, a set of recommended practices at each of the maturity levels, and a maturity assessment model. (Gao, J. Tsao, H & Wu, Y. 2003.356-357)

TABLE 2. Software testing standards

ISO/IEC 9126-1:2001	Software Engineering	Product Quality	Part1: Quality Model
ISO/IEC 14598-1:1999	Information Technology	SW Product Evaluation	Part1: General Overview
ISO/IEC 14598-2:2000	Software Engineering	Product Evaluation	Part2: Planning and Management
ISO/IEC 14598-3:2000	Software Engineering	Product Evaluation	Part3: Process for Developers
ISO/IEC 14598-4:1999	Software Engineering	Product Evaluation	Part4: Process for Acquirers
ISO/IEC 14598-5:1998	Information Technology	SW Product Evaluation	Part5: Process for Evaluators
ISO/IEC 14598-6:2001	Software Engineering	Product Evaluation	Part6: Documentation of Evaluation Modules
ISO/IEC 12119-9:1994	Information Technology	Software Packages	Quality Requirements and Testing
ISO 9127:1988	Information Processing Systems	User Documentation and Cover Information for Consumer Software Packages	

IEEE, Institute of Electrical and Electronics Engineers, has also defined the software testing standards. The key standards according to Edward Kit (1995, 189) are listed in Table 3. Kit (1995, 190) introduces other standards related to testing that are listed in Table 4.

TABLE 3. Key software testing standards

IEEE/ANSI 829:1983	Standard for Software Test Documentation	Defines the content and format of 8 documents that covers the entire testing process
IEEE/ANSI 1008:1987	Standard for Software Unit Testing	Defines an integrated approach to systematic and documented unit testing

TABLE 4. Other standards related to software testing

IEEE/ANSI 1012:1986	Standard for Software Verification and Validation Plans (SVVP)	To provide uniform and minimum requirements for the format and content of SVVP's, To define specific V&V tasks and their required inputs and outputs, To suggest optional V&V tasks to be used to tailor SVVP's
IEEE/ANSI 1028:1988	Standard for Software Reviews and Audits	Provides direction to the reviewer or auditor on the conduct of evaluations.
IEEE/ANSI 730:1989	Standard for Quality Assurance Plans	Establishes a required format and a set of minimum contents for SW quality assurance plans
IEEE/ANSI 610.12:1990	Standard Glossary of Software Engineering Terminology	Promotes clarity and consistency in the vocabulary of SW engineering and associated fields

Test Maturity Model (TMM) is to provide a framework for assessing the maturity of the test processes in an organisation, and providing targets on improving the maturity. TMM is not a standard, but its impact on testing is quite similar when it is used.

There are the following five levels of maturity:

> **Level 1 – Initial** At this level an organisation is using ad-hoc methods for testing, so results are not repeatable and there is no quality standard.
> **Level 2 – Definition** At this level testing is defined a process, so there might be test strategies, test plans, test cases, based on requirements. Testing does not start until products are completed, so the aim of testing is to compare products against requirements.
> **Level 3 – Integration** At this level testing is integrated into a software life cycle, e.g. the V-model. The need for testing is based on risk management, and the testing is carried out with some independence from the development area.
> **Level 4 – Management and measurement** At this level testing activities take place at all stages of the life cycle, including reviews of requirements and designs. Quality criteria are agreed for all products of an organisation (internal and external).

> **Level 5 – Optimisation** At this level the testing process itself is tested and improved at each iteration. This is typically achieved with tool support, and also introduces aims such as defect prevention through the life cycle, rather than defect detection (zero defects). (Testing Maturity Model from Wikipedia, 17.December 2011.)

3.6 Test Engineer

A test engineer should have skills and knowledge to fulfil the qualification of a tester position. Knowledge skills are easy to see as a possession of technological knowledge and management of tools and techniques. Personal qualifications are more difficult to identify and measure. Furthermore, they vary within technologies, companies and projects.

3.6.1 What the Work Requires for a Tester

Tester's assignment is to prevent the bugs and detect them in order to get them handled in an appropriate manner. The bugs appear along the code is written so the developers do "develop" the bugs. Developers have many techniques to prevent bugs; code reviews, pair coding, unit testing, etc. None of these alone nor together prevent all the problems. Software testers can help by providing a dynamic process of executing the software in varying environments with realistic data and with as much input variation as can be imagined in the short cycles in which the testing occurs. (Whittaker, J. 2010, 12-13.)

Testing work is an expense, while writing code is constructive work, testing is destructive. To be worth of funding, your work must improve customer satisfaction and increase your company's profits. (Kaner, C., Falk, J. & Nguyen, H. 1999, 255.)

Testers may participate collecting the requirements, something which designers did alone earlier.

> Because Scrum teams shift focus during requirements gathering from writing about requirements to talking about them, conversations with the product owner become the tester's primary way of finding out how a new feature should behave. A tester is likely to talk with the product owner about how a feature should work, how quickly it should perform, what acceptance criteria must be passed, and so on. Testers are not limited to acquiring this information solely from the product owners. As appropriate, testers should also talk with users, customers, and other stakeholders. (Cohn, M. 2010, 149.)

Testing work seems impossible to do well, because of complex products and incomplete expectations for nebulous quality concerns.

> Software tester wanted. Position requires comparing an insanely complicated, poorly documented product to a non-existent or woefully incomplete specification. Help from original developers will be minimal and given grudgingly. Product will be used in environments that vary widely with multiple users, multiple platforms, multiple languages, and other requirements yet unknown but just as important. We're not quite sure how to define them, but security and performance are paramount, and post release failures are unacceptable and could cause us to go out of business. (Whittaker, J. 2010, 21.)

If the reality was this devastating, who would bother to do testing work? Testing is regardless of its destructive approach meaningful work and there the results are important.

In 1995, Beizer had some ideas how this importance could be higher. He listed his hopes for the future of testing as follows:

1. That testing becomes a standard part of the software developer's under graduate education. As mandatory part at least three different levels in an undergraduate course of study: Black-Box testing, Integration and System Testing, Testing Theory and Algorithms.
2. That it keeps pace with our ever-evolving software development process and apparently ever-increasing software complexity.
3. The test tool industry takes its rightful place as an essential component of the broader software development tools industry.
4. That for most of us, testing ceases to be a profession, but an inseparable aspect of what every conscientious developer routinely does. (1995, 244.)

3.6.2 Certified Skills

The ISTQB Foundation level syllabus qualifies that a certified tester remembers and understands the fundaments of the testing and testing process, SW development models, test levels, test types and test techniques as well as test tool types, risks and testing, test progress monitoring and control. A certified tester is capable of applying and analyzing the testing techniques, test planning and estimation, test management and incident management.

The ITIL Foundation level syllabus describes the learning objectives as follows: candidates can expect to gain knowledge and understanding in the following upon the successful completion of the education and examination components related to this certification.

- Service management as a practice (comprehension)
- The ITIL service lifecycle (comprehension)
- Generic concepts and definitions (awareness)
- Key principles and models (comprehension)
- Selected processes (awareness)
- Selected functions (awareness)
- Selected roles (awareness)
- Technology and architecture (awareness)
- Competence and training (awareness)

> CAT, the Certified Agile Tester certificate, has a lifelong validity. There, no re-certification is required. The examinees will gain an understanding of the testing role within an agile project and be able to effectively apply practical skills associated with that role. At the end of the course, successful students will be able to:
>
> -Understand the principles behind the agile approach to software development
>
> -Differentiate between the testing role in agile projects compared with the role of testers in non-agile projects
>
> -Positively contribute as an agile team member focused on testing
>
> - Appreciate the challenges and difficulties associated with the non-testing

activities performed in an agile team

- Demonstrate a range of soft skills required by agile team members

(Certificates – iSQI – Certifying People all over the world, 19. November 2011.)

The purpose of the certification is to gain the sufficient level of knowledge. It is a way to recognize professional achievement, provide a career path, and introduce an incentive to learn about testing and related fields. It is important to note that the certification is not a registration or a license. (Craig, R. & Jaskiel, S. 2006, 326-327.)

3.6.3 Definitions and Descriptions for a Good Tester

What kind of people makes a good tester? What are those qualifications? They have a balanced curiosity to chase down problems, an understanding not to perform unnecessary investigation and a willingness to have an extra mile when required without tormenting themselves too tired to work. (Black, R. 2002, 276-279.)

Participants of Test Manager Classes have listed the characteristics of good testers as such:

- Is inquisitive
- Has functional/business knowledge
- Is detail oriented
- Is open-minded
- Has a good personality
- Has a technical background, but does not want to be a programmer
- Has testing experience
- Is a team player
- Is flexible
- Is self-reliant
- Is self-starting
- Has a positive attitude
- Is logical
- Handles stress well
- Is a quick thinker
- Knows specific tools
- Has good common sense
- Is politically astute
- Has sense of humor

- Understands the software development lifecycle
  (Craig, R. & Jaskiel, S. 2006, 312.)

The listed characteristics are dependent on the speaker. The management values different characteristics than the testers. All the skills are not needed at every level of the test and most probably all of them do not even exist in one person. Even at different development models have diversified the qualifications for testers, as an agile tester.

> A tester provides information to stakeholders about the software being developed. A tester helps customer define functional and non-functional requirements and quality criteria, and helps turn these into tests that guide development and verify desired behaviour. Testers perform a wide variety of activities related to delivering high-quality software, such as test automation and exploratory testing. In agile development, everyone on the development team performs testing activities. Team members who identify themselves as tester work closely with other members of both the developer and customer teams. (Crispin, L. & Gregory, J. 2009, 499.)

Important principles for an agile tester are as follows:

- Provides continuous feedback.
- Delivers value to customer.
- Enables face-to-face communication.
- Has courage.
- Keeps it simple.
- Practices continuous improvement.
- Responds to change.
- Is self-organizing.
- Focuses on people.
- Enjoys.

(Crispin, L. & Gregory, J. 2009, 22.)

Some features for agile testers come straight from the Agile Manifesto, for example "individuals and interactions over process and tools"- principle encourages team members to communicate. Testers on a Scrum team will have to become accustomed to more frequent and meaningful conversations with their coworkers and, in many cases, people outside the team. (Cohn, M. 2010, 149.)

> It is the job of the tester to report the failures and discrepancies observed to the author and/or to the management. The manner of this reporting can contribute the cooperation between developers and testers, or have a negative influence on the important communication of these two groups. To prove other people's errors is not an easy job and requires diplomacy and tact.(Spillner, A. , Linz, T. & Schaefer, H. 2007, 31.)

As a team-member, a tester still has a special role in the team. An Agile tester on the agile team provides early feedback during all stages of development, helps or is cognizant of the code-level testing being performed, takes the lead on acceptance test automation building, regression test plans and uncovers additional test scenarios through exploratory testing. (Carter, J. 26.November 2011.)

Testers are the gate keepers for quality and an information source for risk management. Testers must understand that their business role is to evaluate the business risk and to report those results to the management. (Perry, W. 2000, 8.)

Team members will interact in ways they see modeled by those they consider leaders in the organization or to their team, including product owners, any functional managers to whom the team members report, and other executives and managers in the organization. To foster the right kind of behavior, then, team and organizational leaders should demonstrate the type of learning behaviors they would like to see on their teams. (Cohn, M. 2010, 210.)

3.6.4 Employers' Requirements

Employers have also added some personal skills to their open position advertisements. Some open position advertisements are included as appendices to this thesis and the requirements for testers are the following:

Test Environment Engineer (Appendix 1)

Test Engineer/ Specialist (Appendix 2)

Test Engineer/Test Automation Engineer (Appendix 3)

3.7 Quality Assurance

A dictionary defines Quality Assurance as a program for the systematic monitoring and evaluation of the various aspects of a project, service, or facility to ensure that standards of quality are being met. (Merriam-Webster-dictionary, 13.November 2011.)

Testing of software contributes to the improvement of the software quality, but it entails more than just the elimination of failures that occurred during the testing. (Spillner, A. , Linz, T. & Schaefer, H. 2007, 10.) The definition of the testing quality is conducted from the same sources than quality definitions for software development in general. The CMMI (Capability Maturity Model Integration) model is widely used to implement Quality Assurance in an organization.

The customer's perception of the quality is the one that matters in the end.
If the customer does not like the end result, it does not matter if the product meets a specification, even if the customer has agreed to the specification. For that customer, it is not good quality if she is not happy with it. (Kaner, C. , Falk, J. & Nguyen, H. 1999, 59.)

There are different possible ways to gain what is believed to be the customer's perception of the quality. Testing is quality control. Quality assurance means preventing bugs. This objective is unachievable because of the challenge of the increased complexity that the users want. Users keep escalating their quality expectations, thereby stretching the testing ability and techniques. (Beizer, B. 1995, 13.)

Quality is not only related to the product under development, but to the organization and used methods for development. As well the methods and practices for achieving the best quality for a reasonable price are chosen. 'Agile' is not a practice. It is a quality of the organization and its people to be adaptive, responsive, continually learning and evolving – to be *agile,* with the goal of competitive business success and rapid delivery of economically valuable products and knowledge. (Larman, C. , Vodde, B. 2008, 139.)

Quality belongs to all, and all who have some impact on the project are responsible of quality. Because the team members are on the same team and everyone works on the same user stories at the same time, the team seems to have more unity. Before using Scrum, it seemed each function (analyst, programmer, tester, DBA) was done in a silo. (Cohn, M. 2010, 139.) Even when the quality belongs to everybody the responsibility of the quality does not spread evenly. Testing engineers in agile teams have more influence on the development process and, what is even more important, on the end product. (Cohn, M. 2010, 151.)

3.8 Test Automation

An increased emphasis on test automation becomes a hallmark of Scrum teams. Even teams that have struggled for years to make progress in automating tests find that the short sprints of Scrum make test automation a necessity. Over time this reduces the reliance on manual testers: those who read a script, push a button, and note the results. These testers often find themselves being asked to learn one or more of the test automation tools used by the team. (Cohn, M. 2010, 149.)

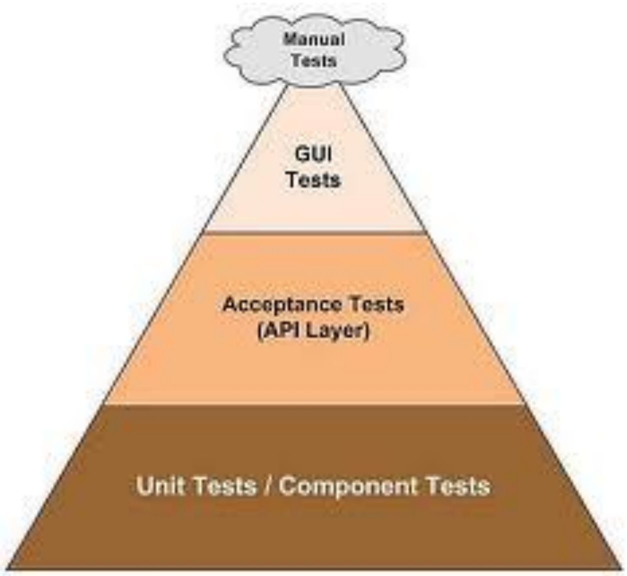


FIGURE 10. Test Automation Pyramid

Figure 10 illustrates the three different layers of automated tests. The foundation, which supports all layers above it, consists of the Unit- and Component tests. After a team has mastered the art of TDD (Test Driven Development), these tests are by far the quickest and least expensive to write. (Crispin, L. & Gregory, J. 2009, 276.) Also, because unit tests are usually written in the same language as the system, programmers are often most comfortable with writing them. (Cohn, M. 2010, 312.)

The middle layer of the pyramid includes most of the automated business-facing tests written to support the team. These ‘story tests’ or ‘acceptance tests’ verify that we are “building the right thing”. (Crispin, L. & Gregory, J. 2009, 277.) Mike Cohn draws the same pyramid with different layer names (see Figure 11).The service describes the target of the testing, and the testing the service API’s.   (Cohn, M. 2010, 312.)

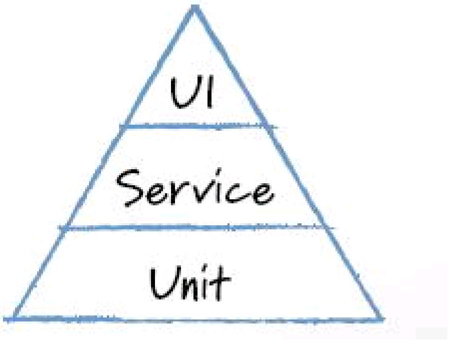


FIGURE 11. Test Automation Pyramid by Cohn

These acceptance tests need their own framework for maintaining and execution. Therefore, the used coding language can be very independent on the actual software code language, and the test responsible code designers may be as well testers as programmers.

Involving the programmers in finding cost-effective ways to automate the top-level GUI tests has multiple benefits. These efforts may give the programmers a better understanding of the system’s ”big picture,” and testers can learn how to create mobile pliable, less straw-like GUI tests. (Crispin, L. & Gregory, J. 2009, 279.) Tool selection is the key for a successful GUI automation. The automated scripts have to be flexible and easy to maintain. Making the code easy to maintain increased the ROI on these tests. (Crispin, L. & Gregory, J. 2009, 283.)

Some types of the testing cannot be done without automation. Manual load tests are not usually feasible or accurate, and performance testing requires both monitoring tools and a way to drive actions in the system under test. (Crispin, L. & Gregory, J. 2009, 283.)

Automation is not only for testing. There are tasks that have to be repeated time after time all over again; automated scripts can replace these repetitive tasks.

3.9 The Role of Manual Testing

It is impossible to fully automate all tests for all environments. Further, some tests are prohibitively expensive to automate. Many tests that cannot be chosen to automate involve hardware or integration to external systems. Exploratory testing should feature short, feedback-generating cycles through these steps in a manner analogous to the test-driven development's short cycle of the test-code-refractor. Beyond finding bugs, exploratory testing can also identify missing test cases. These can then be added at the appropriate level of the test automation pyramid. Further, exploratory testing can uncover ideas that are missing from the user story as initially understood. It can also help a team discover resolutions that seemed like a good idea at the time but seem like bad ideas now that the feature has been developed. These situations usually result in new items being added to the product backlog. (Cohn, M. 2010, 314.)

Manual testing is the best choice for finding bugs related to the underlying business logic of an application. Business logic is the code that implements user requirements; in other words, it is the code what for the customers buy the software. Business logic is complex and requires a human in the loop. (Whittaker, J. 2010, 14.)

Many non-functional tests are impossible to automate; maintainability, scalability, testability, usability. Real usability testing requires a real user to use the test object under test. Observing users in action, debriefing them on their experiences, and judging the results is a job for a person who understands that usability aspects of software cannot be automated. (Crispin, L. & Gregory, J. 2009, 285.)

In some occasions, automation is so expensive or even impossible to establish that manual testing is the only option. When the software target is an individual person to use, it is best to test the user interface manually as there is no automation to judge the usefulness of it.

4 DISCUSSION

As the status of the tester in software projects has changed over time so has the career path. The previously valid roles for testers are not much in use anymore. Today the agile tester has only one role as an agile team member with sufficient skills from the previous roles. If the agile team acquires a tester as a quality assurance, the base of the skills has to be expanded from the previous testing roles.

4.1 What Has Changed?

Software testing has transformed from the debugging code to a separate phase of developing work. First, the testing started after the implementation work was finished, then it was divided to different testing types to follow different development phases, and eventually it became merged with developing in an agile developing team. Along the way a whole career path was developed for a tester with several dedicated positions in the testing area, which seemed to vanish as obsolete in the agile world.

During this change the tester has learned to study the requirements to be able to interpret them to the test cases and to judge the right problems from the wrong ones. The standards guide the testers' work as much as they do for developers, and the tester needs to be aware of them. To stay qualified for a tester job a tester studies to meet the certification requirements. As software products become more and more complex, a tester has to manage more techniques and tools to do the testing properly. While a tester executes the testing task, s/he is capable of producing a whole set of reports, a tester is a professional user of table, text and presentation expressions. A tester can give reconstructive feedback to developers and does not get upset if somebody has a better understanding of the testing work. A tester knows the cons and pros for the test automation and is capable of utilizing automation wherever it is feasible. S/he learns to create and maintain the acceptance test code to help the programmers to code the wanted features, as they are required. Even when a tester also becomes a programmer as s/he is when writing the code for testing, s/he is keeping the tester's attitude and does not become exposed to programmer's point of view.

The main task of the current tester is to provide quality information from different stakeholder’s points of views. As testing is context driven, there is no one matching method, no all covering tool, not one good strategy: a tester has to be flexible and use all s/he knows and study more.

4.2 Who Defines the Roles and Needed Skills for Testers?

Every organization defines the roles and the positions for their work, nevertheless these seldom come from scratch, and usually they are copied from previous projects or companies. Certificates describe and define some roles in software development and they are widely used as a description of the position. People assume to know what kind of responsibilities these roles express.

The idea of being a certified professional is to maintain and deepen the knowledge, not forget everything once the certification is accomplished. The level of knowledge is also depending on the candidate’s experience; a rookie has studied the knowledge but has not lived it true. A certification is the way to recognize a professional achievement, provide a career path, and introduce an incentive to learn about testing and the related fields, it implies that the requirements have been fulfilled to become certified and that an exam has been passed. (Craig, R. & Jaskiel, S. 2006, 326-327.) -The Finnish testing professionals association, “Testaus OSY- FAST”, has a conversation that calls out the universal certification which could undoubtedly prove the level of knowledge. The employer’s frustration can be understood when a certified employee cannot apply the certified knowledge in work.

Certifications ensure that a certified tester remembers and understands or is aware of certain models, practices, processes, techniques, and is capable of applying and analyzing plans, techniques, estimations and management of testing and incidents or comprehends the principles and models. All this comes from books and prints. A certified tester has at least once read or heard about the questions which are generally rated as essential to the work as a tester.

IEEE standards provide a detailed description of the unit tests and documentation format. (Gao, J. Tsao, H & Wu, Y. 2003, 357.) The purpose of the standards is to guide projects to achieve their goals successfully by giving documented instructions how it is feasible to do. The Agile manifesto emphasizes a working software over the documentation. The Agile framework also emphasizes the use of the most suitable methods and tools each time and the change of them whenever it is needed. The institutions that grant certifications have multiple reasons how the certification holder benefits it.

Other side of the coin is that certificates are also business activity and these certifications don’t come for free. A candidate has to arrange time and some enthusiasm to participate the training course and pass the exam. Both parts have their fees.

These certificates belong to institutes and do not change rapidly, as they have to be general and global. An applicant may not get any tools to resolve the current situation by studying a certificate. A good side is that certificates give general knowledge of the area and are useful despite the development method in use. Anyway, the invested money for the certificates should be considered carefully case by case.

As long as testing fundamentals do not belong to the basic studies, the software companies have to decide how they can get skilled people to do the testing work. Arranging the needed training is expensive, whether it happens inside the company and / or purchased from special training companies.

Testing certifications try to assure that applicants do have the common knowledge of the testing fundamentals. This is a respectable goal, but not every tester can achieve these certifications. Companies train their employees for their own purposes; does the induction have to start from the basics?

Different roles defined for software testers are listed in Table 5.

TABLE 5. Different defined roles for testers

	Roles	**No roles**
ISEB/ISTQB certificates	Hierarchical career roles	
CAT certificate		Special skills
SCRUM	3 different team roles	
KANBAN		Team members

4.3 What Kind of People Software Development Needs Now?

A tester obtains a wide understanding about the software s/he is working with and a tester understands the business values of the features and is capable of prioritizing the risks for the product. A tester is capable of communicating face-to-face and written text with the customer and is able to interpret the customer's needs to programmers, so that together they can produce value for the customer.

A tester does not let anybody influence his/her work attitude; s/he keeps his/her mind open while focusing on people. A tester does not need anybody to tell him/her what to do; s/he is able to see what has to be done within a team. A tester practices continuous improvement with testing, by himself / herself and with the development team. Meanwhile, s/he is responsive to changes and enjoys the development work.

It would be easy if it was so simple. There are many different limitations directing the work, and many of them come along with the size and the structure of the organization. Some intend to keep the chosen development method as a limit for the work processes.

Not every team or even software development department has the luxury of having a complete control over its development process. For example, in outsourced contract development, customers often mandate the suppliers to be the CMMI Level 5 certified, which requires those software developers to follow certain established best practices. Additionally, some software-intensive products are delivered into regulated industries and must comply with standards such as ISO 9001. The companies producing medically regulated devices must comply with ISO 13458. None of these standards prescribe a life cycle that is completely at odds with Scrum. (Cohn, M. 2010, 425.)

Most of the cases it is merely rethinking what people need when they see something that inhibits for working effectively and innovatively. Processes are there for keeping the work in shape and ensuring the repeatable quality. When a process does not fulfill this expectation, it becomes obsolete.

The quality process has three distinct goals: improving a software product, assessing the quality of the software product, and improving the long –term quality and cost-effectiveness of the quality process itself.(Pezze, M & Young, M. 2008,12) Improvement requires innovativeness. A company can hire innovative people, but if the company's culture does not support that innovativeness, it will die. In addition, hired innovativeness may be addressed to something which does not benefit the company at all. Innovativeness must be nourished and let flourish, supported and maintained, which is not an easy task.

> What can make the process of innovation more predictable? It does *not* entail learning to predict what *individuals* might do. Rather, it comes from understanding the *forces* that act upon the individuals involved in building business – forces that powerfully influence what managers choose and cannot choose to do. (Christensen, C. & Raynor, M. 2003, 9.)

If innovativeness is a character of a person, the management is the key actor to lure the innovativeness to benefit the company and the project. Innovativeness can be one of the company values, but companies still need actions to keep it as a part of the company culture. Even inside the company, in different projects the innovativeness can vary from top to bottom.

The capabilities of management of the testing group are critical, as is the manager of the testing manager. Since testing is performed in a separate organization from development, it takes extra effort to encourage and create a positive team environment. (Kit, E.; edited by Finzi, S. 1995, 169)

A positive team environment ensures that people stay within the team, but there are some fundamental skills which helps the team work as a team. One of them is how the team develops during its existence. This knowledge has nothing to do with software development but only team development in general. It is common team behaviour.

4.4 How to Find the Testers?

How to find good testers for the job? Either hire new graduates and train them, or look to the other sources such as development and QA departments, or poach them from other companies' testing departments. (Craig, R. & Jaskiel, S. 2006,313.) The hiring process is challenging as such, but emphasizing the right attitude on applicants makes it pure art. Each team member has to have a strong focus on quality and delivering the business value. "Consider more than just technical skills when you're expanding your team." (Crispin, L. & Gregory, J. 2009, 69.) Craig recommends defining the job requirements first, to know what is needed and to recognize it when it is there. "If you're not finding that person after several interviews, be open to re-evaluate your job requirements and seeing what parts can be modified". (Gao, J. Tsao, H & Wu, Y. 2003, 21.) Larman and Vodde advise to hire the best, and if the best is not found, avoiding the hiring. (2008, 280-281.)

When hiring these development team members with testing capabilities, their roles and careers have to be reconfigured in the human resources department aligned to their real work. Testing should not be considered only an inevitable expense, but also an activity that brings value to the product. Testing is an essential part of the software development work and quality assurance. If the code itself was all the value, there would not be testing at all.

How different authors point the essential aspects of hiring testers are listed shortly in Table 6.

TABLE 6. Different aspects of hiring testers

	YES	NO
Hiring from other companies test departments	Craig, R. & Jaskiel, S. 2006,313	
Consider also other than only technical skills	Crispin, L. & Gregory, J. 2009, 69 Cohn, M. 2010, 149 Perry, W. 2000, 8	Whittaker, J. 2010, 12-13
Hire only the best	Larman, C. , Vodde, B. 2008, 280-281	Gao, J. Tsao, H & Wu, Y. 2003, 21

5 CONCLUSION

This thesis problem was obvious; the change in software testing was about to happen. The difficult part was to find out where the change will lead, and this thesis work is one interpretation at this time. In some years, the outcome may become very different.

A tester's role has changed from a bug hunter to the bug preventer. As a bug preventer, the impact on the developed software is bigger than ever before. The more the tester's position goes towards quality assurance, the more the tester is responsible for the whole software value.

Testers also need reverence for their work and developed skills. When this does not come along the project roles, there should be some other way, and the companies have to discover how to show the appreciation to their employees. As testers are gatekeepers for the quality of the product and their work adds value to the product, should they gain some value too? What inspires the testers to continue working as team members time after time? The testers should find the appreciation from their skills, abilities and knowledge. The company's Human Resource department should support this appreciation to come visible.

Testing is context-driven, and testers provide information about the test object to the stakeholders, and information is different regarding the stakeholder's aspect to the test object. This approach gives room for innovative thinking and continuous improvement in the testing branch in the future as well.

Further research could concentrate on clarifying what kind of testing is most profitable from the most satisfied customer's point of view.

REFERENCES

An Encyclopedia Britannica Company; Merriam-Webster- Dictionary
Available at: http://www.merriam-webster.com/dictionary/quality+assurance
Date of acquisition 13.November 2011.

Bach,J. , Kaner, C. & Pettichord, B. 2002. Lessons learned in Software Testing . USA: John Wiley & Sons, Inc.

BCS Certifications
Available at: http://certifications.bcs.org/
Date of acquisition 26. November 2011.

Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, Dave Thomas. Agile Alliance::The Agile Manifesto.
Available at: http://www.agilealliance.org/the-alliance/the-agile-manifesto/
And at: http://agilemanifesto.org/
And at: http://agilemanifesto.org/principles.html
Date of acquisition 19. January 2012.

Beizer, B. 1995. Black-Box Testing. USA: John Wiley & Sons, Inc.

Belt, P. 2009. Improving verification and validation activities in ICT companies- product development management approach, University of Oulu

Black, R. 2002. Managing the Testing Process. USA: Wiley Publishing, Inc.

Carter, JoEllen The Agile Tester
Available at: http://www.versionone.com/pdf/WP_AgileTester.pdf
Date of acquisition 26.November 2011.

Certificates – iSQI – Certifying People all over the world
Available at: https://www.isqi.org/en/certificates.html
Date if data acquisition .19 November 2011.

Christensen, C. & Raynor, M. 2003. The Innovators Solution. Boston: Harvard Business School Press

Cohn, M. 2010. Succeeding with Agile Software Development using Scrum . USA: Addison- Wesley.

Craig, R. & Jaskiel, S. 2006. Systematic Software Testing 5th edition. USA: Artech House Publishers

Crispin, L. & Gregory, J. 2009. Agile Testing. Boston: Pearson Education, Inc .

Gao, J. Tsao, H & Wu, Y. 2003. Testing and Quality Assurance for Component-based Software. Norwood: Artech House, Inc.

Hezel, B. 1988. The Complete Guide to Software Testing. John Wiley and Sons.

History of BCS| Role and Purpose
Available at: http://www.bcs.org/category/11284
Date of acquisition 26. November 2011.

Home - ISTQB International Software Testing Qualifications Board
Available at: http://istqb.org/display/ISTQB/About+Us?atl_token=b6-m_sUiQn
And at: http://istqb.org/display/ISTQB/Certification?atl_token=gUOwRFDwx8
Date of acquisition 15. November 2011.

Kaner, C. , Falk, J. & Nguyen, H. 1999.Testing Computer Software 2nd edition. USA: John Wiley & Sons, Inc.

Kit, E.; edited by Finzi, S. 1995. Software Testing in the Real World. Great Britain:ACM Press

Kniberg, H. & Skarin, M. 2010. Kanban and Scrum- making the most of both. USA: C4Media Inc.

Larman, C. , Vodde, B. 2008. Scaling Lean & Agile Development . USA: Addison-Wesley.

Meerts, Joris. Testing References
Available at: http://www.testingreferences.com/testinghistory.php
Date of acquisition 26. November 2011.

Myers, Glenford J. 1979.*The Art of Software Testing*. John Wiley and Sons.

Perry, W. 2000. Effective Methods for Software Testing 2nd edition. USA: John Wiley & Sons, Inc.

Pezze, M & Young, M. 2008. Software Testing and Analysis. USA: John Wiley & Sons, Inc.

PPM Studio::Agile Project Management Software for Agile software development
Available http://www.ppmstudio.com/Agile-Software-Development.aspx
Date of acquisition 19. January 2012.

Schwaber, K. 2007. The Enterprise and Scrum. Washington: Microsoft Press.

Spillner, A. , Linz, T. & Schaefer, H. 2007. Software Testing Foundations 2nd edition. Santa Barbara: Rocky Nook Inc.

The Official ITIL® Website
Available at: http://www.itil-officialsite.com/AboutITIL/WhatisITIL.aspx
And at: http://www.itil-officialsite.com/Qualifications/ITILQualificationScheme.aspx,
Date of acquisition 15. November 2011.

Testing Maturity Model From Wikipedia
Available at: http://en.wikipedia.org/wiki/Testing_Maturity_Model
Date of acquisition 17.December 2011.

The Homepage of The Association for Software Testing
Available at: http://www.associationforsoftwaretesting.org/about/governance/
And at: http://www.associationforsoftwaretesting.org/about/
Date of acquisition 26. November 2011

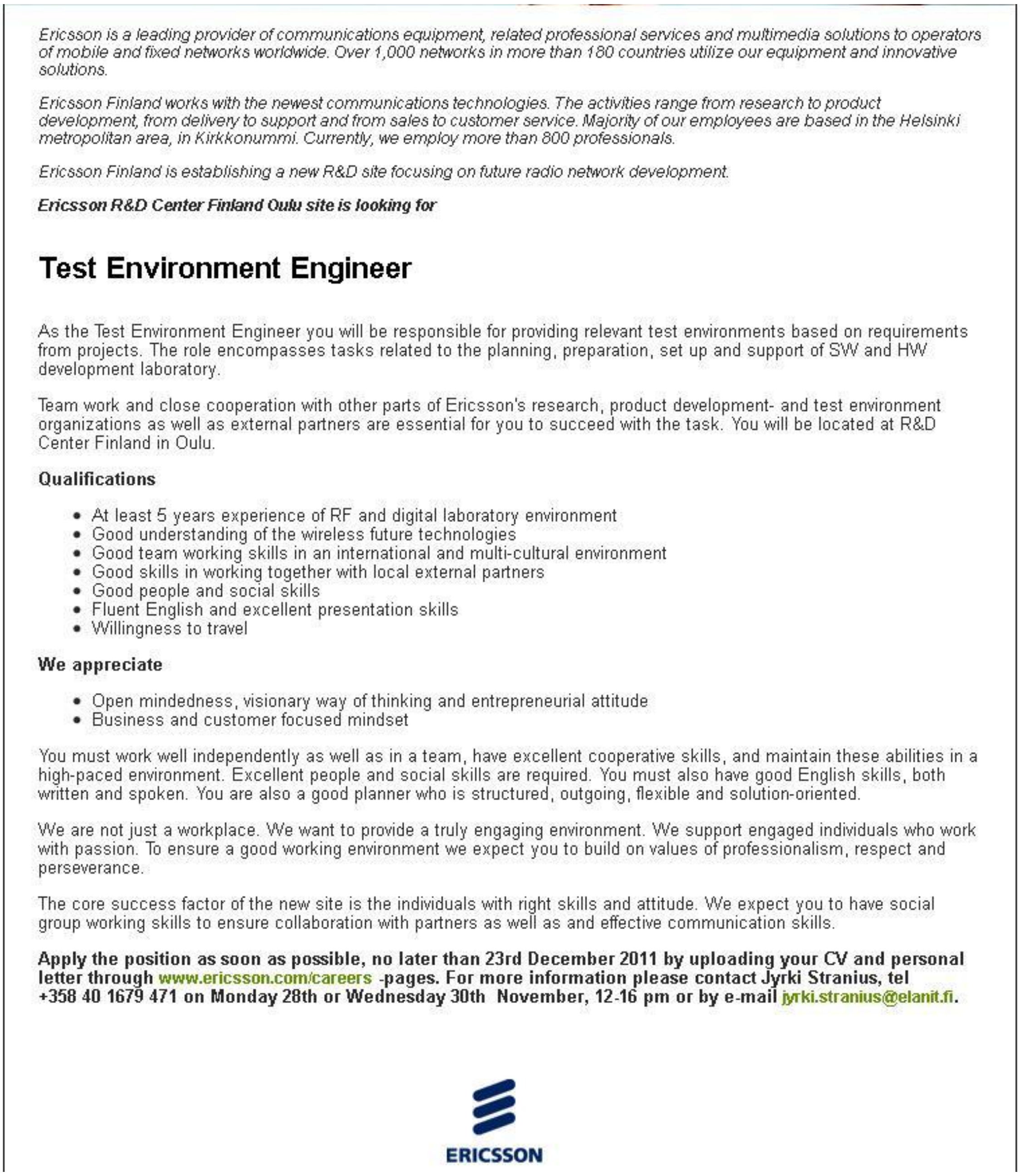
The Official ITIL® Website
Available at: http://www.itil-officialsite.com/AboutITIL/WhatisITIL.aspx
And at: http://www.itil-officialsite.com/Qualifications/ITILQualificationScheme.aspx,
Date of acquisition 15. November 2011.

V-Model From Wikipedia, the free encyclopedia,
Available at: http://en.wikipedia.org/wiki/V-Model
Date of data acquisition 19 January 2012.

Waterfall-model From Wikipedia, the free encyclopedia,
Available at: http://en.wikipedia.org/wiki/Waterfall_model
Date of data acquisition 19 January 2012.

Whittaker, J. 2010. Exploratory Software Testing. Boston: Pearson Education, Inc .

Test Environment Engineer Appendix 1



Ericsson is a leading provider of communications equipment, related professional services and multimedia solutions to operators of mobile and fixed networks worldwide. Over 1,000 networks in more than 180 countries utilize our equipment and innovative solutions.

Ericsson Finland works with the newest communications technologies. The activities range from research to product development, from delivery to support and from sales to customer service. Majority of our employees are based in the Helsinki metropolitan area, in Kirkkonummi. Currently, we employ more than 800 professionals.

Ericsson Finland is establishing a new R&D site focusing on future radio network development.

Ericsson R&D Center Finland Oulu site is looking for

Test Environment Engineer

As the Test Environment Engineer you will be responsible for providing relevant test environments based on requirements from projects. The role encompasses tasks related to the planning, preparation, set up and support of SW and HW development laboratory.

Team work and close cooperation with other parts of Ericsson's research, product development- and test environment organizations as well as external partners are essential for you to succeed with the task. You will be located at R&D Center Finland in Oulu.

Qualifications

- At least 5 years experience of RF and digital laboratory environment
- Good understanding of the wireless future technologies
- Good team working skills in an international and multi-cultural environment
- Good skills in working together with local external partners
- Good people and social skills
- Fluent English and excellent presentation skills
- Willingness to travel

We appreciate

- Open mindedness, visionary way of thinking and entrepreneurial attitude
- Business and customer focused mindset

You must work well independently as well as in a team, have excellent cooperative skills, and maintain these abilities in a high-paced environment. Excellent people and social skills are required. You must also have good English skills, both written and spoken. You are also a good planner who is structured, outgoing, flexible and solution-oriented.

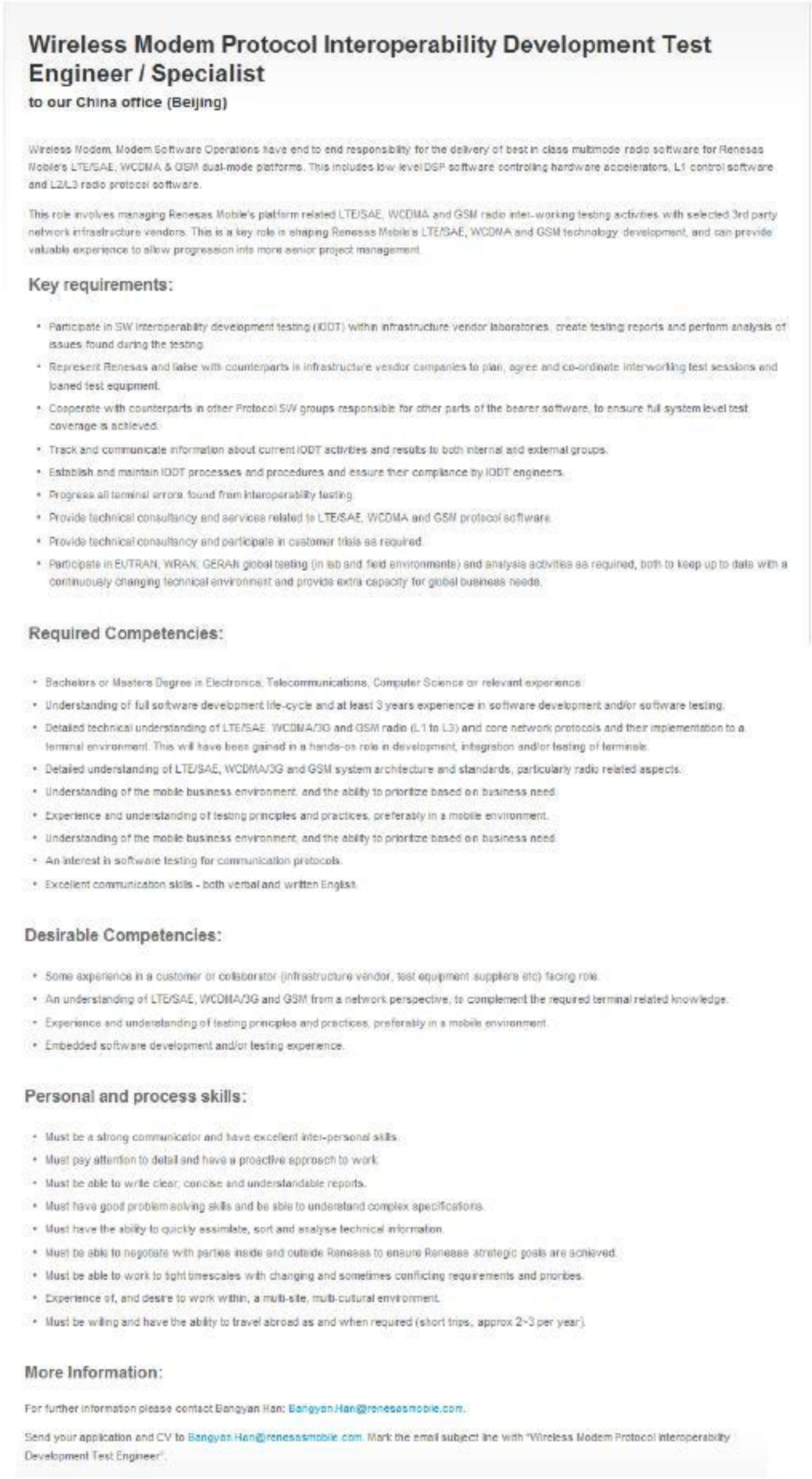
We are not just a workplace. We want to provide a truly engaging environment. We support engaged individuals who work with passion. To ensure a good working environment we expect you to build on values of professionalism, respect and perseverance.

The core success factor of the new site is the individuals with right skills and attitude. We expect you to have social group working skills to ensure collaboration with partners as well as and effective communication skills.

Apply the position as soon as possible, no later than 23rd December 2011 by uploading your CV and personal letter through www.ericsson.com/careers -pages. For more information please contact Jyrki Stranius, tel +358 40 1679 471 on Monday 28th or Wednesday 30th November, 12-16 pm or by e-mail jyrki.stranius@elanit.fi.

ERICSSON

Test Engineer/ Specialist Appendix 2



Wireless Modem Protocol Interoperability Development Test Engineer / Specialist

to our China office (Beijing)

Wireless Modem, Modem Software Operations have end to end responsibility for the delivery of best in class multimode radio software for Renesas Mobile's LTE/SAE, WCDMA & GSM dual-mode platforms. This includes low level DSP software controlling hardware accelerators, L1 control software and L2/L3 radio protocol software.

This role involves managing Renesas Mobile's platform related LTE/SAE, WCDMA and GSM radio inter-working testing activities with selected 3rd party network infrastructure vendors. This is a key role in shaping Renesas Mobile's LTE/SAE, WCDMA and GSM technology development, and can provide valuable experience to allow progression into more senior project management.

Key requirements:

- Participate in SW interoperability development testing (IODT) within infrastructure vendor laboratories, create testing reports and perform analysis of issues found during the testing.
- Represent Renesas and liaise with counterparts in infrastructure vendor companies to plan, agree and co-ordinate interworking test sessions and loaned test equipment.
- Cooperate with counterparts in other Protocol SW groups responsible for other parts of the bearer software, to ensure full system level test coverage is achieved.
- Track and communicate information about current IODT activities and results to both internal and external groups.
- Establish and maintain IODT processes and procedures and ensure their compliance by IODT engineers.
- Progress all terminal errors found from interoperability testing.
- Provide technical consultancy and services related to LTE/SAE, WCDMA and GSM protocol software.
- Provide technical consultancy and participate in customer trials as required.
- Participate in EUTRAN, WRAN, GERAN global testing (in lab and field environments) and analysis activities as required, both to keep up to date with a continuously changing technical environment and provide extra capacity for global business needs.

Required Competencies:

- Bachelors or Masters Degree in Electronics, Telecommunications, Computer Science or relevant experience.
- Understanding of full software development life-cycle and at least 3 years experience in software development and/or software testing.
- Detailed technical understanding of LTE/SAE, WCDMA/3G and GSM radio (L1 to L3) and core network protocols and their implementation to a terminal environment. This will have been gained in a hands-on role in development, integration and/or testing of terminals.
- Detailed understanding of LTE/SAE, WCDMA/3G and GSM system architecture and standards, particularly radio related aspects.
- Understanding of the mobile business environment, and the ability to prioritize based on business need.
- Experience and understanding of testing principles and practices, preferably in a mobile environment.
- Understanding of the mobile business environment, and the ability to prioritize based on business need.
- An interest in software testing for communication protocols.
- Excellent communication skills - both verbal and written English.

Desirable Competencies:

- Some experience in a customer or collaborator (infrastructure vendor, test equipment suppliers etc) facing role.
- An understanding of LTE/SAE, WCDMA/3G and GSM from a network perspective, to complement the required terminal related knowledge.
- Experience and understanding of testing principles and practices, preferably in a mobile environment.
- Embedded software development and/or testing experience.

Personal and process skills:

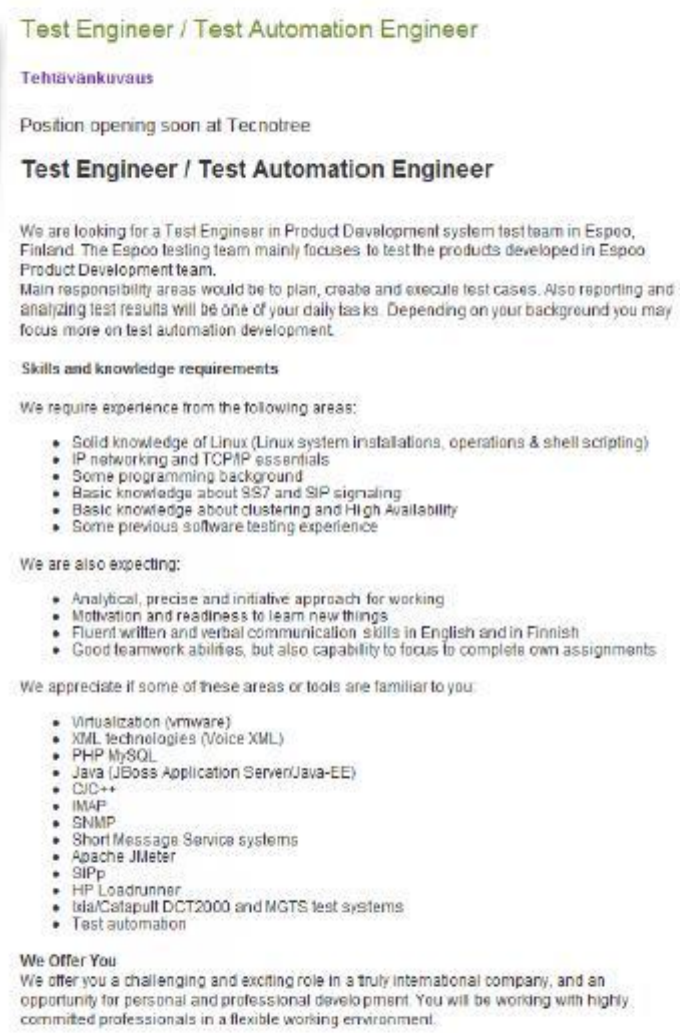
- Must be a strong communicator and have excellent inter-personal skills.
- Must pay attention to detail and have a proactive approach to work.
- Must be able to write clear, concise and understandable reports.
- Must have good problem solving skills and be able to understand complex specifications.
- Must have the ability to quickly assimilate, sort and analyse technical information.
- Must be able to negotiate with parties inside and outside Renesas to ensure Renesas strategic goals are achieved.
- Must be able to work to tight timescales with changing and sometimes conflicting requirements and priorities.
- Experience of, and desire to work within, a multi-site, multi-cultural environment.
- Must be willing and have the ability to travel abroad as and when required (short trips, approx 2~3 per year).

More Information:

For further information please contact Bangyan Han: Bangyan.Han@renesasmobile.com.

Send your application and CV to Bangyan.Han@renesasmobile.com. Mark the email subject line with "Wireless Modem Protocol Interoperability Development Test Engineer".

Test Engineer/Test Automation Engineer Appendix 3



Test Engineer / Test Automation Engineer

Tehtävänkuvaus

Position opening soon at Tecnotree

Test Engineer / Test Automation Engineer

We are looking for a Test Engineer in Product Development system test team in Espoo, Finland. The Espoo testing team mainly focuses to test the products developed in Espoo Product Development team.
Main responsibility areas would be to plan, create and execute test cases. Also reporting and analyzing test results will be one of your daily tasks. Depending on your background you may focus more on test automation development.

Skills and knowledge requirements

We require experience from the following areas:

- Solid knowledge of Linux (Linux system installations, operations & shell scripting)
- IP networking and TCP/IP essentials
- Some programming background
- Basic knowledge about SS7 and SIP signaling
- Basic knowledge about clustering and High Availability
- Some previous software testing experience

We are also expecting:

- Analytical, precise and initiative approach for working
- Motivation and readiness to learn new things
- Fluent written and verbal communication skills in English and in Finnish
- Good teamwork abilities, but also capability to focus to complete own assignments

We appreciate if some of these areas or tools are familiar to you:

- Virtualization (vmware)
- XML technologies (Voice XML)
- PHP MySQL
- Java (JBoss Application Server/Java-EE)
- C/C++
- IMAP
- SNMP
- Short Message Service systems
- Apache JMeter
- SIPp
- HP Loadrunner
- Ixia/Catapult DCT2000 and MGTS test systems
- Test automation

We Offer You
We offer you a challenging and exciting role in a truly international company, and an opportunity for personal and professional development. You will be working with highly commited professionals in a flexible working environment.