Minna Tiitinen

Key Factors for Selecting Software Testing Tools

Helsinki Metropolia University of Applied Sciences

Master Degree in Business Informatics
Key Factors for Selecting Software Testing Tools

Date 30.11.2013

Abstract

Author(s) Title Pages Date	Minna Tiitinen Key factors for selecting software testing tools 101 pages + 3 appendices 30th November 2013
Degree Programme	Master Degree of Business Informatics
Instructor(s)	Thomas Rohweder, DSc (Econ), Principal Lecturer Kari Kakkonen, Director, Testing and Methodologies, Päivi Brunou, Senior Testing Consultant

This study presents information of software testing, importance of testing, testing tools selection process and consequences of serious defects in software. A key element to conduct successful software testing, are various testing tools. In this study software testing tools are categorised according to ISTQB classification. In total seven classifications were included into this study. Most popular software testing tools and their usage are presented different acquiring methods of software testing tools; commercial, open source, proprietary and cloud service tools are also covered. The testing tool selection process is complicated. Information will be given from the traditional selection process for software testing tools and test automation tools.

The object of the study was to give general information about software testing and related tools and make a survey of used testing tools for software and IT-professionals based on the background investigation for Knowit Oy. The main objective of this thesis was to investigate the use of software testing tools in Finland, produce reports and presentations and finally present some recommendations for the selection process of the testing tools, including information on which functionalities and other issues are affecting the selection process. One object of the results from this study, is to direct marketing efforts towards most used testing tools and their training.
This study is limited to selected 127 software testing tools. The main source for survey respondents came from selected Know it's customer base with software testing activities and a TestausOSY mailing list.

As a result of this study two confidential reports and one public report were generated. The confidential reports; in English and in Finnish, are for Knowit to distribute within customers. The public report was published on the web page of Knowit and presented in Knowit's morning seminar and Testing Days 2013 in Otaniemi, Espoo. Recommendations for testing tools selection process, life-cycle models and tool developments are also included in the study. Finally, conclusions and the summary of the study are presented.

Keywords	Software testing, tools, automation, quality, tool classification, tool selection, life-cycle model, testing methods, survey, cloud

ACKNOWLEDGEMENTS

For the second time I have studied now a degree in the evenings, besides a job. First degree was a Bachelor of Engineering Technology at the Helsinki Polytechnic, 2004. Studying in the evenings is hard, but it is very rewarding in the end. I believe that the life is a lifetime learning process. And I am very grateful for this opportunity. They say a rolling stone gathers no moss.

First, I want to thank Principal Lecturer, DSc (Econ) Thomas Rohweder from Helsinki Metropolia, University of Applied Sciences, who has been my instructor for the thesis. He has done tireless work for guiding and instructing me through the process. I also thank Anne-Mari Raivio for the support and instructing with the English language.

I also extend my gratitude to Knowit Oy who made this study possible. Especially, I want to thank my instructors Kari Kakkonen for good ideas, comments and professionalism and Päivi Brunou for guiding through the thesis content. I also thank Ville Särmälä, Jani Taipaleenmäki, Markus Kääriäinen, Tiina Henry-Biabaud, Sari Eronen-Mäkelä and Minna Aalto for their help and support. I also want to thank my team leader Nina Perta and colleagues who have made comments and supported both my studies and thesis.

Last, but not least I deeply appreciate the support and understanding of my family because of my almost impossible schedules and absence from home.

In the end my sincere thanks to my school mates and colleagues and teachers of the Metropolia for inspiring and motivating atmosphere during the studies towards a Master’s Degree in Business Informatics.

Minna Tiitinen

Espoo, November 2013.

Contents

1 Introduction 1

1.1 Background 1
1.2 Knowit Oy 5
1.3 Objective 5
1.4 Delimitations 7

2 Research process 7

2.1 Flowchart of the process 7
2.2 Data collection and analysis methods 10

3 Best practises of software testing and software testing tools 10

3.1 What is software testing? 11
3.1.1 Used life-cycle models in software testing 12
3.1.2 Importance of life-cycle selection 19
3.1.3 Importance of testing 20
3.1.4 Why is quality important for a business? 21
3.1.5 Examples of consequences of serious defects 22
3.1.6 What is a software testing tool? 25
3.1.7 Benefits of using testing tools 27
3.1.8 Risks in using software testing tools 27
3.1.9 What is automated software testing? 28
3.1.10 Why to automate? 29
3.2 Testing tool classification 31
3.2.1 Management tools 31
3.2.2 Execution tools 33
3.2.3 Static testing tools 35
3.2.4 Performance testing tools 37
3.2.5 Test specs tools 39
3.2.6 Specific tools 39
3.3 Methods selecting testing tools traditionally 40
3.3.1 Software testing tool evaluation 41
3.3.2 Open source tools 45
3.3.3 Custom tools 46
3.3.4 Tools selection process 47

3.3.5 Automation testing tool evaluation 48
3.3.6 Evaluation of vendors 49
3.4 Summary 53

4 Conducting the survey 55

4.1 Planning of the survey 56
4.1.1 Target group of the survey 56
4.1.2 Questions of the survey 57
4.1.3 The main content of the survey 57
4.1.4 Selected software testing tools 59
4.1.5 Sending the survey 59
4.1.6 Reports taken from the database 61
4.2 Analysis of the survey 61
4.2.1 Analysis of general questions 61
4.2.2 The current and future life-cycle models 64
4.2.3 The most popular testing tools according to the survey 66
4.2.4 Purpose of testing tools 69
4.2.5 Testing tools acquiring methods 70
4.2.6 Do the companies use office tools for testing? 72
4.2.7 Popularity of open source tools vs. commercial tools? 73
4.2.8 What would be the requirements for a new testing tool? 75
4.2.9 Training of software testing professionals 76
4.2.10 Evaluation of most popular testing tools and vendors 77
4.2.11 Analysing the comments from the respondents 78
4.3 Results 81
4.3.1 Public report 81
4.3.2 Reports to customers 81
4.3.3 Presentations 82

5 Development plan 83

5.1 Proposal for life-cycle models 83
5.2 Methods to acquire software testing tools 83
5.3 Recommendations for selecting software testing tools 83
5.4 Recommendations for software testing tool development 85

6 Conclusions 86

6.1 Summary and discussion 86

6.2 Practical next steps 88
6.3 Evaluation of the study 89
6.3.1 Outcome/ Objective 89
6.3.2 Qualitative and quantitative research 90
6.3.3 Validity and reliability 92

References 94

Appendices

Appendix 1. Testing tool survey questions
Appendix 2. Most popular testing tools in the Finland – public report
Appendix 3: Testing tool selection list Lewis, W.E., (2004)

ABBREVIATIONS/ACRONYMS

Agile	Agile methodology is an alternative to traditional project management, typically used in software development.
Agile Manifesto	The Agile Manifesto, also called the Manifesto for Agile Software Development, is a formal proclamation of four key values and 12 principles to guide an iterative and people-centric approach to software development.
ALM	Application Life-cycle Management
Cloud service	It is an IT provisioning and support model that provides on-demand network access to a shared pool of computing resources (Spirent, 2010).
Digium Enterprise application	Digium Enterprise is an Internet-based research and data collection service.
Debugger	A special program used to detect errors (bugs) in other programs. A debugger allows a programmer to stop the program at any point and examine and change the values of variables.
GUI	Graphical User Interface
ICT	Information and Communications Technology
IID	Iterative and Incremental Development
ISTQB	International Software Testing Board. ISTQB® has created the world's most successful scheme for certifying software testers. As of March 2013, ISTQB® has issued over 295,000 certifications in 70 countries world-wide, with a growth rate of approximately 12,000 certifications per quarter. The scheme relies on a Body of Knowledge (Syllabi and Glossary) and exam rules that are applied consistently all over the world, with exams and supporting material being available in many languages.
IT	Information technology
Kanban	Kanban is a Lean method for managing the creation of products with an emphasis on continual delivery while

	not overburdening the development team. Kanban is a process designed to help teams work together more effectively. Kanban is based on 3 basic principles: • Visualize what you do today (workflow): seeing all the items in context of each other can be very informative. • Limit the amount of work in progress (WIP): this helps balance the flow-based approach so teams don’t start and commit to too much work at once • Enhance flow: when something is finished, the next highest thing from the backlog is pulled into play
Lean	“Lean” is the set of management practices based on the Toyota Production System (TPS). One way of defining Lean: Eliminate waste and non-value-added activity (NVA) through continuous improvement and practice respect for people.
Pace-layered application strategy	Gartner's Pace-Layered Application Strategy is a new methodology for categorizing applications and developing a specialised management and governance process that reflects how they are used and their rate of change.
ROI	Return on investment
Scrum	Software life-cycle method. Scrum is the most popular way of introducing Agile due to its simplicity and flexibility.
SDPM	Software Development Process Model
Soliditet	Soliditet is a Business Area and an auxiliary trade name of Bisnode Finland Oy. Bisnode Group is the leading provider of Credit Information and digital Business Information in Europe. www.soliditet.fi

Survey	The survey is a series of tasks that finally results a statistical file of statistical units and their characteristics (variables).
SWEBOK	Software Engineering Body of Knowledge. It is an international standard ISO/IEC TR 19759:2005 specifying a guide to the generally accepted Software Engineering Body of Knowledge.
TA	Test Automation
TestausOSY	Finnish association of software testing. www.testausosy.fi
Time-boxed	Period with limits, a limited period during which an action, process, or condition exists or takes place.
V-model	The V-Model is a product-development process initially established in Germany for government defence projects. It has become a mutual standard in software development. The V-Model obtains its name from the fact that the process is regularly outlined as a flow diagram that acquires the form of the letter V.
Waterfall method	The waterfall model is a sequential design process, often used in software development processes, in which progress is seen as flowing steadily downwards (like a waterfall) through the phases of Conception, Initiation, Analysis, Design, Construction, Testing, Production/Implementation, and Maintenance.

LIST OF FIGURES

Figure 1; Software testing process (Kaushalam, Information Technology architect)
Figure 2; Classification of selected tools
Figure 3; Flowchart of the process
Figure 4; Testing activities (Iconma. Testing methodology)
Figure 5; Waterfall model by The Smart Method Limited
Figure 6; V-model defined by gss
Figure 7; Factors on challenged projects (Johnson, J. et al. 1994).
Figure 8; Agile method explained by gss
Figure 9; Scrum process by cPrime
Figure 10; Distribution of the used testing tools over software types (Khaled M. M. et al, 2009).
Figure 11; Software testing tools in different development phases. (Fewster et al. 1999)
Figure 12; Different types of performance testing. (Buksh, J. 2011)
Figure 13; The tool selection and implementation process. (Fewster et al. 1999)
Figure 14; Magic Quadrant for Integrated Software Quality Suites 2013. (Gartner 2013)
Figure 15; Software testing mind map by author
Figure 16; Welcoming -page of the survey
Figure 17; Survey has sent to 510 contacts
Figure 18; Age of the respondents
Figure 19; Operating area of the company
Figure 20; Acquiring method of the software testing tools
Figure 21; Current life-cycle model in the project
Figure 22; Future life-cycle models
Figure 23; Test management tools used among software testers
Figure 24; Overall usage of software tools
Figure 25; Purpose of the software testing tools
Figure 26; The acquiring methods of software testing tools
Figure 27; Participation in the tool selection process vs. role in the company
Figure 28; So called “office tools” are used same functions as testing tools
Figure 29; Used software testing tools

Figure 30; Acquiring method of the software testing tools

Figure 31; Functionalities affecting to testing tool selection process

Figure 32; Respondents’ training

Figure 33; Evaluation of most popular tools

LIST OF TABLES

Table 1; Typical examples of software defects (McDonald, M., et al, 2007)

Table 2; Example of costs fixing defects (Rothman, J. 2000)

1 Introduction

1.1 Background

Today, computers and applications are integral part of everyday living; almost everybody has one laptop or more, cell phones, tablets and pocket calculators at home. Often a computer is controlling machines, although people do not recognize that the system is including a computer. For example; valves, sensors and controlling systems are controlled by the machine. These kinds of systems are called embedded systems. They can be very small units as in cell phones or significant ones as in aircrafts. Since computers are an integrated part of everyday life, applications and computer programs need to be tested more effectively. If for example, applications used in hospitals were not tested carefully, there could be very serious consequences. Therefore, not all the testing can be done manually, instead automation and testing tools are used to assist in testing.

Software Testing is a process consisting of test life-cycle activities including both static and dynamic testing. These life-cycles consist of phases such as planning, preparation and evaluation of software products. The software products determine satisfaction of customer’s requirements and they are fit for customer’s use. Software testing is carried out to detect software faults or failures in advance. Testing is helped by various software testing tools.

Software testing tools support one or more test activities, such as planning and control, specification, building initial files and data, test execution and test analysis as seen in Figure 1; Software testing process (Kaushalam, Information Technology architect). Test management most commonly refers to the activity of managing the computer software testing process. (ISTQB 2012)

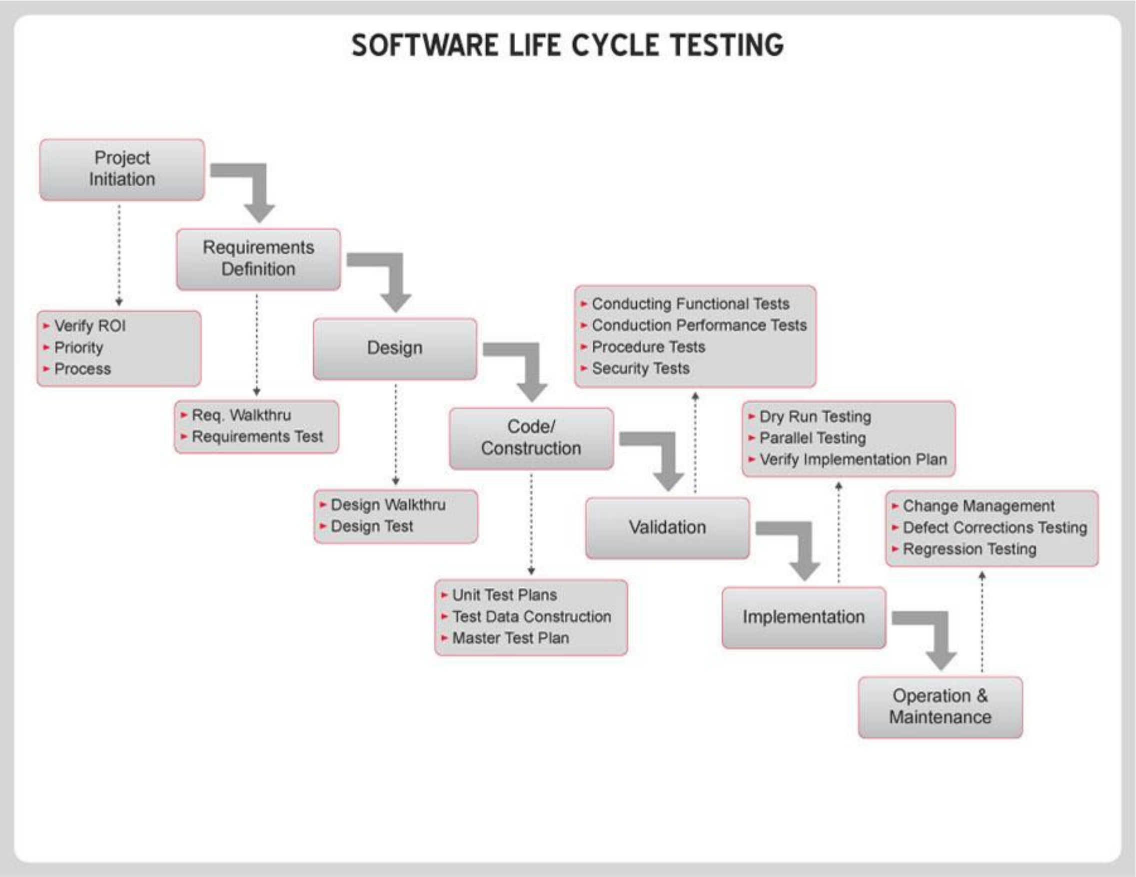


Figure 1; Software testing process (Kaushalam, Information Technology architect)

Test automation is the process of utilizing a computer program to execute a procedure or user transactions against an IT system. This is typically accomplished by utilizing an automated test tool. Automated testing is commonly practised in functional regression testing, performance testing, load testing, network testing and security testing. Automation is very helpful in the projects, where is plenty of regression testing, but has to remember that planning and creating the test scripts is also taking time. If the application changes the script has to update too.

The software testing project can run with several life-cycle methods and new ones are invented all the time. For example Waterfall model, Incremental, V-model, Scrum and Iterative model. Each process model follows a specific life-cycle to ensure success in the procedure of software development. Traditional very popular life-cycle model has been V-model, but agile methods have become more popular. Most popular examples from Agile method is Scrum. Agile methodology is an alternative to traditional project management, typically used in software development. Agile software development is a collection of software development methods based on iterative and incremental development (ISTQB, Testing throughout the testing life-cycle 2012).

In this study different testing tool acquiring methods are evaluated and compared. Traditionally software testing tools are purchased, but other acquiring methods are available too including open source tools, proprietary tools or using cloud services. Cloud services have started internationally become more popular acquiring method.

There are plenty of different applications in use in software testing. Some of them solve some needs in testing, but the perfect software testing tool is not yet invented. Selecting a testing tool or a test management tool for a company is challenging task. Various requirements are needed in the testing tool to fulfil expectations of the users. Especially in the test management tools this is realised, because of the nature of a number of functionalities. The best tools are often also very expensive and smaller companies (customers) cannot buy expensive licenses. It is considered as too significant investment. The best possible testing tool is often a compromise of selection. The company has to decide whether to buy expensive licenses or if the open source application is beneficial enough. (Glenford J. Myers 2004)

Today, over 600 different software testing tools are on the market. It is impossible to include all of them to one study. For this study a careful process of testing tool classification selection was done. This study is following the ISTQB testing tools classification. The tools are grouped by the testing activities or areas that are supported by a set of tools, for example, tools that support management activities and tools to support static testing. Ten tool classifications categories were included in this study. The selected tools and classifications were categorised as following way;

- Test management tools; 18
- Incident management tools; 6
- Static analysis tools; 21
- Test data preparation tools; 7

- Test execution tools; 19
- Test design tools; 11
- The test harness / unit test framework tools; 17
- Specialized unit test framework, 10
- Dynamic analysis tools; 13
- Performance/ load/ stress testing tools; 13

In the end 127 different testing tools were selected to the survey questions.

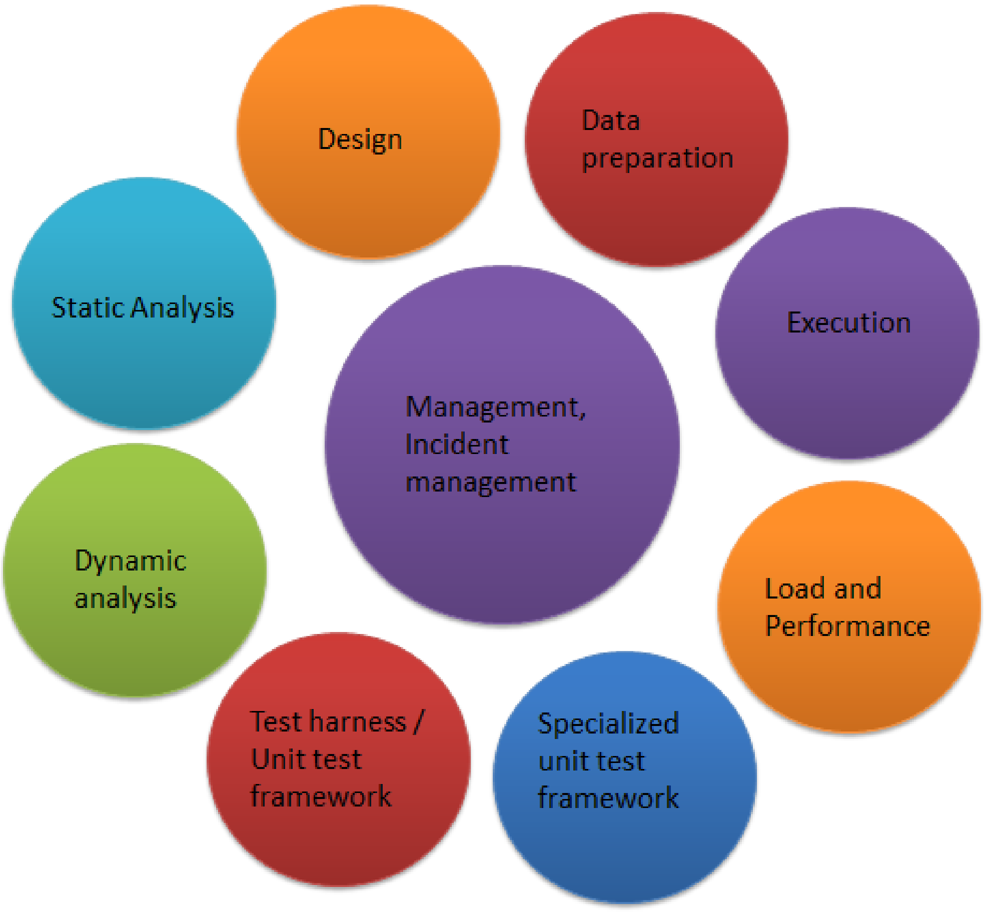

Design
Data preparation
Static Analysis
Execution
Management, Incident management
Dynamic analysis
Load and Performance
Test harness / Unit test framework
Specialized unit test framework

Figure 2; Classification of selected tools

The limits between the groups are out of focus because there are many tools, which can belong to several classes.

The survey was designed based on previously presented information. The survey questions were planned carefully to keep the survey short enough and not too time consuming. After the survey questions were planned, the survey was designed with the aid of a Digium Enterprise tool and sent to selected target groups. The target group was testing professionals from Knowit’s customer database and TestausOSY testing society.

The results of the survey were saved to the database. Because Digium Enterprise tool usage is limited the results was taken to MS Excel spreadsheet. The results were analysed and graphs were created. There was planned one public report and two reports for Knowit for customer purposes.

1.2 Knowit Oy

Knowit is listed on the OMX Stock Exchange and is an ICT service company providing solutions for digital businesses, information management, and software development and testing. The company employs 1,700 IT professionals in the Nordic countries; Finland, Sweden, Norway and Russia. The office of Knowit Finland is responsible for Knowit's operations in Finland and Russia. In Finland Knowit is known for its expertise in digital communications and testing. Knowit seeks to ensure customers’ success via long-lasting partnerships and developing its customers’ competitiveness. Knowit ICT solutions are business-oriented and comprehensive and implemented using state-of-the-art technology. (Knowit Oy 2012)

1.3 Objective

Conducting a study about test tool selection has been a current topic in Knowit Oy for some time. Customers have indicated that testing tool selection is challenging process and asked for recommendations. Testing consultants face daily various testing tools at customer premises. Upon starting a new project Knowit testing consultants have to know or learn several tools quickly. The Knowit testing team has been looking for information about the testing tool selection to help customers and consultants as well as to target testing tool training, but research about the exact tools used in Finland was not available. Knowit Oy is also a training company offering training for both customers and ICT professionals. Internal trainings are organised regularly to keep employees up-to-date. This study, helps concentrating training efforts to most used testing tools, ensuring the up-to date knowledge of testing tools. Results from this study can also be used by the marketing department. For example, listing information of most used testing tools for customer and offering training for customers.

These reasons gave me the opportunity to do my master's thesis about this topic.

The main objective of this thesis is to investigate the use of software testing tools in Finland, produce reports and presentations and finally present some recommendations for the selection process of the testing tools. Only a certain selection of tools was included in the study because it is impossible to go through all of them. In total 127 different tools were selected for the survey.
The main objective of the study is:

- Information of used testing tools is very valuable when Knowit's customers are asking for the best testing tools in practice.
- The Knowit and its' consultants can give information of the most used software testing tools based on the survey results and information about the strengths and weaknesses of different tools.
- Information about the most used testing tools affect what kind of training to give to testing consultants and the Knowit's customers.
- In addition, Knowit's marketing professionals can utilise the survey information in marketing work.
- Recommendations on how to select testing tools based on the survey and which functionalities and other issues are affecting the selection process.

The thesis will produce altogether three reports; one in Finnish and two in English showing graphs of the survey results and analysis. Knowit Oy will deliver these reports in two ways;

1. To selected customers are given detailed level information in English report
2. To selected customers are given detailed level information in Finnish report
3. Survey respondents are given summarized information and public report published on the internet page of Know It Oy.

The idea was to study the latest information of software testing tools by reading the articles and books of the subject. Secondly the survey was planned and sent to the customer database of Knowit Oy, Solidiet register of Finland and Testing OSY mailing list. It was the idea to bring a wide range of responses from different industries. The obtained data was saved to the database. Finally, the results were analysed and some recommendations made. Research methods used was quantitative method. The study is based on the information found from literature shown in the reference list, at the end of the study.

1.4 Delimitations

This study is limited to selected software testing tools, altogether 127 tools. There are available hundreds of software tools and it is not possible to go through all of them. For example mobile-computing and database testing tools are excluded from this study.

Several methods to classify testing tools are also available. For Example ISO 9126 and SWEBOK (Software Engineering Body of Knowledge). This study follows the ISTQB classification method of software testing tools.

There is plenty of different software development life-cycle in use. This study is presented only Waterfall, V-model, Agile and Scrum software life-cycle models.

Basic sample for the survey has selected among Knowit’s customers having software testing activities. The survey is sent only to

- Testing customers of the Knowit
- Soliditet register of Bisnode Finland
- TestausOSY (Finnish Association of software testing) mailing list

The study concentrates only questions related to testing and software testing tools. The survey should keep short as possible that respondents do not get tired to respond to the questions. The survey included already 61 questions and was quite long, but still many interesting issues had to leave out of the survey.

In testing tools classification was followed the classification of the ISTQB (International Software Testing Board).

2 Research process

2.1 Flowchart of the process

This section presents the overall process of research; identifying the business problem and objective, planning the research problem, searching for best practises e.g. of literature, implementing the survey, analyse the results and finally produce a development plan based on the survey results.

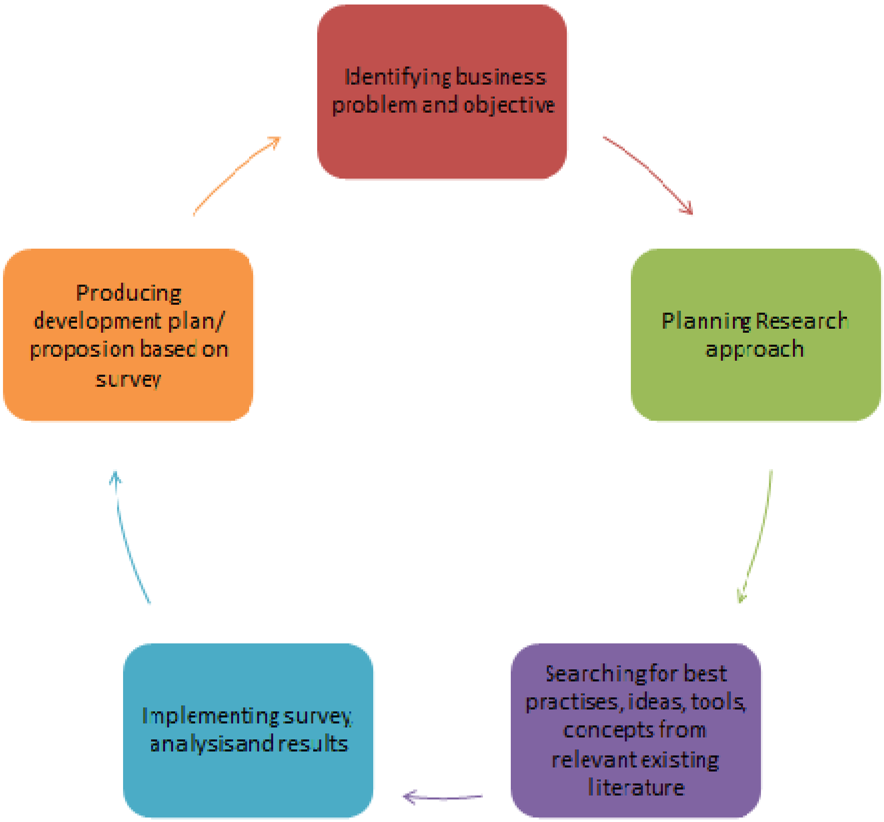

Identifying business problem and objective
Planning Research approach
Searching for best practises, ideas, tools, concepts from relevant existing literature
Implementing survey, analysis and results
Producing development plan/ proposion based on survey

Figure 3; Flowchart of the process

Identifying the business problem and objective, the topic of the research is software testing tool usage in Finland. The Knowit has been thinking for a long time to produce a research about the topic, but there has been a lack of resources to do that. The training and marketing department of the Knowit can utilise this information further. The results give idea to which direction to take the training. The marketing department can prove to customers that the company owns the latest knowledge of the used testing tools.
The research is answering the following questions:

- What are software testing tools and why testing is important?
- What are the most popular software testing tools in use?
- Who is using the testing tools?
- What life-cycles models are used in the companies?
- What are the future trends in life-cycle models?

- Do the companies use only one tool or several testing tools?
- How popular are open source tools compared to commercial tools?
- Are users happy to current tools?
- What functionalities are missing from the current tools?
- What requirements there are for testing tools?
- What would be the requirements for new testing tool?
- Recommending how to select testing tools

For the research approach a survey to Knowit’s customer base was selected. The survey was first drafted in MS Word-document and then transferred into Digium Enterprise software. After that the survey was mailed to the Knowit customers emails via Digium Enterprise application. The survey was also sent to Testaus OSY community mailing list. TestausOSY is Finnish Association of Software Testing. (TestausOSY)

There are several surveys and research generally made of software testing, but not so many of exactly the software testing tools. Background information was searched and investigated whether there are any other surveys found on the subject. Finally, the found information was investigated, sorted and selected the best articles and books on the subject.

The survey planning started studying how to design a survey. After that the designing of the survey questions started, based on the literature. Questions were reviewed by thesis instructor several times and some modifications were made. Also my colleagues contributed with comments. After reviewing the survey was sent for testing to a few colleagues. Eventually, the survey was sent out by web form and email.

Finally, there was analysed phase of the outcome and was designed several graphs. Results were presented following way;

- First time in at Breakfast seminar of Knowit, in the Kämp meeting room, on 26 of May in 2013.
- The poster was planned and implemented and it was presented in Testing Days 2013, in Otaniemi, on the 4th of June 2013.

- Later, public version of the results was published on the internet page of Knowit Oy.
- Finally, two confidential reports in Finnish and in English were planned and published to the Knowit's customer purposes. The final phase of the analysis was to produce a development plan based on the survey; how to select testing tools for a company.

2.2 Data collection and analysis methods

The survey planning started with finding out generally how the survey will be designed and what are the best practises of it. Professional of Digium Enterprise user was consulted about planning and structure of the survey and possibilities of the design tool. Knowit had promised to draw a movie theatre ticket package as a prize for one of the respondents.

The survey was structured into following parts:

- Email cover sheet
- Survey first page
- General questions
- Questions related to testing
- Testing tools questions
- Participation to the draw

As mentioned before, the survey was sent by Digium Enterprise application and TestausOSY mailing list on 21st of February. Time for answering was about two weeks. The reminder was sent on 5th of March and the survey was closed on 8th of March. The results of the survey were saved to database and taken out to the MS Excel spreadsheet. Finally, the analysis by MS Excel pivot tables started.

3 Best practises of software testing and software testing tools

This section covers areas of software testing, the importance of it and the importance of quality. In addition, examples of serious software defects will be presented.

3.1 What is software testing?

Software testing is one of the key processes in delivery of applications. Very common though is that software testing is only running the tests, for example executing software after software source code has been developed. It is true that this is a part of testing, but not the only testing activity. Software testing is more properly viewed as the destructive process of trying to find the errors in a program. A successful test case causes the program to fail. Testing is performed to uncover and correct as many potential errors as possible before delivery to the customer. (Glenford J. Myers 2004, Murphy, T.E. 2013)

The definition of testing according to the ANSI/IEEE 1059 standard is that “testing is the process of analysing a software item to detect the differences between existing and required conditions (that is defects/errors/bugs) and to evaluate the features of the software item”.

Testing activities include a wide range of activities before and after test execution. See the Figure 4; Testing activities (Iconma. Testing methodology) (ISTQB 2012, ISTQB Certification – Foundation Level syllabus)

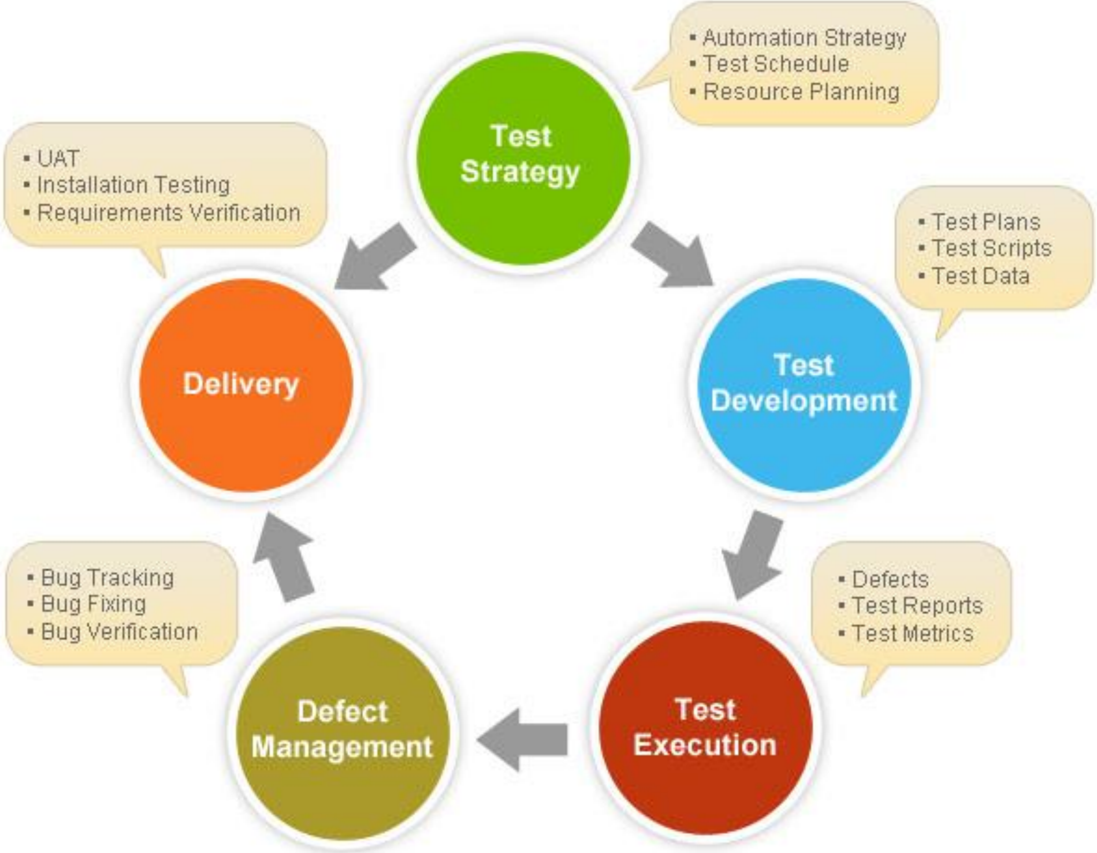


Figure 4; Testing activities (Iconma. Testing methodology)

Testing can have the following objectives;

- To ensure that the solution meets the business and user requirements
- To catch errors that can be bugs or defects
- Idea to determine user acceptability
- To ensure that a system is ready for use
- To gain confidence that program works
- To show that a system performs as intended
- To verify the documentation. (Khannur, A. 2011)

The overall main objective of software testing is to check:
1. That the application is working as expected without any errors or bugs (Functionality)
2. The performance of the application is as expected and meets the need (ISTQB 2011).

Designing test cases early in the life-cycle and selecting correct testing processes and activities can help the project to prevent defects existing in the code. An importance of verifying test basis against test design should be realised. Testing documents reviewing also helps prevent defects in the code (ISTQB 2011, *Expert level Syllabus - Test management)*

Different objectives in testing have to take into account as in the development phase of testing have to cause as many failures as possible in order to fix them in very early phase of testing. The main objective in acceptance testing is to obtain the confidence of the system and that it meets the requirements (ISTQB 2011, *Advanced level Syllabus - Test Analyst)*

3.1.1 Used life-cycle models in software testing

There are various software development methodologies identified and figured which are used throughout the development process of software. These methodologies are called Software Development Process Models. For example, Waterfall model, Incremental, V-model, Agile, Iterative model, RAD and Spiral model. Each process model follows a specific life-cycle to ensure success in the procedure of software development. In this chapter is presented only Waterfall, V-model, Agile and Scrum which is a sub-model for Agile. (ISTQB, Testing throughout the testing life-cycle 2012).

A software life-cycle model illustrates stages of the software cycle and the order in which those stages are performed. Each stage produces deliverables needed by the next stage in the life-cycle. Requirements are transformed into design. Code is created according to the design which is called development stage. After coding and development has implemented, the testing confirms the deliverable of the implementation stage against requirements. In general these six stages are present in every Software development life-cycle model:

- Requirement gathering and analysis
- Design
- Implementation or coding
- Testing
- Deployment
- Maintenance (ISTQB, Testing throughout the testing life-cycle 2012)

3.1.1.1 Waterfall model

The Waterfall life-cycle model was one of the most used models in 1970's. In this development model code was written and after that debugged. The system was not officially designed and the quality criteria was not possible to check. The waterfall model is an example of a plan-driven process—in principle, you must plan and schedule all of the process activities before starting work on them (Bhuvaneswari et al. 2013). Different phases of Waterfall model are shown in Figure 5; Waterfall model by The Smart Method Limited.

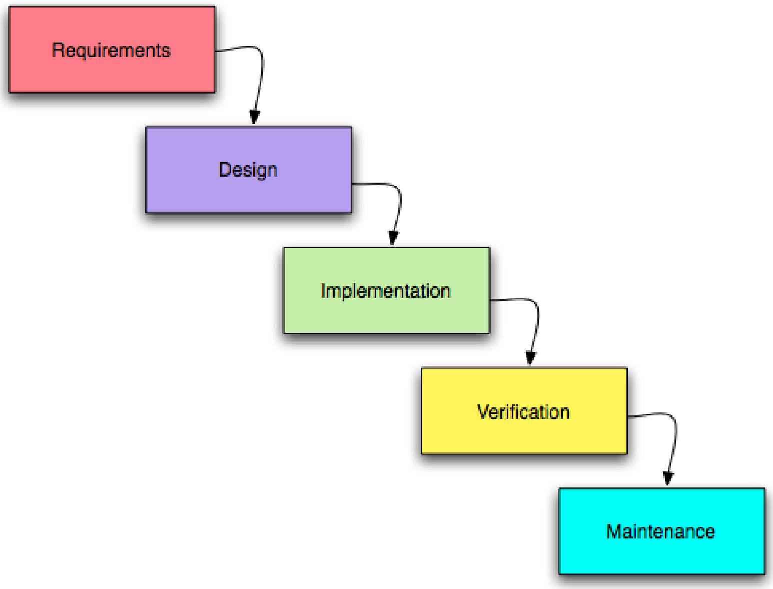


Figure 5; Waterfall model by The Smart Method Limited

Advantages of Waterfall method were;

- Easy to understand and practical to implement
- It supports helpful practises such as defined before designed, designed before coded.
- It characterizes deliverables and milestones,
- Document driven method
- Published documentation standards
- Operates well on mature products and weak teams. (Bhuvaneswari et al. 2013)

Disadvantages:

- It does not replicate the iterative nature of exploratory development.
- It is not realistic to wait for precise requirements so early in the project.
- Software is provided later in the project,
- Delays discovery of serious errors.
- It is difficult to participate in risk management.
- It is expensive to make changes to documents.
- Because of the costs of producing and approving documents, development can be costly and involve significant rework. (Bhuvaneswari et al. 2013)

3.1.1.2 V-model

The V-Model is a product-development process initially established in Germany for government defence projects. It has evolved into a common standard in software development. The V-Model obtains its name from the fact that the process is regularly outlined as a flow diagram acquiring the form of the letter V. (Rouse, M. 2013)

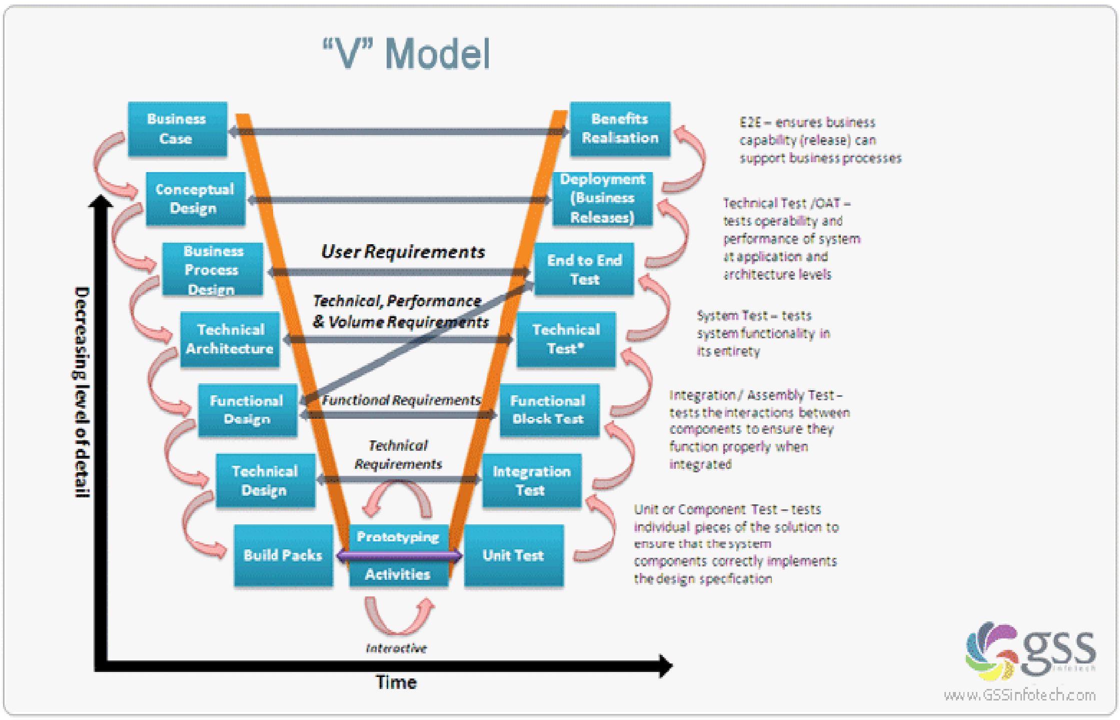

"V" Model
Business Case
Conceptual Design
Business Process Design
Technical Architecture
Functional Design
Technical Design
Build Packs
Prototyping
Activities
Unit Test
Integration Test
Functional Block Test
Technical Test*
End to End Test
Deployment (Business Releases)
Benefits Realisation
User Requirements
Technical, Performance & Volume Requirements
Functional Requirements
Technical Requirements
Interactive
Decreasing level of detail
Time
E2E – ensures business capability (release) can support business processes
Technical Test /OAT – tests operability and performance of system at application and architecture levels
System Test – tests system functionality in its entirety
Integration / Assembly Test – tests the interactions between components to ensure they function properly when integrated
Unit or Component Test – tests individual pieces of the solution to ensure that the system components correctly implements the design specfication
gss
www.GSSinfotech.com

Figure 6; V-model defined by gss

There are plenty of challenges in Waterfall and V-model. The Waterfall life-cycle method drives several high-risk and difficult elements towards the end of a project. While Iterative and Incremental Development methods (IID), run by risk-driven iterations, take up and work out the hardiest and the riskiest elements early. Waterfall works on the scale of three or six weeks in the iteration or in a short project. The collapse happens as complexity grows, rates multiply and feedback is delayed. Waterfall requires less changes, less novelty, and less complexity issues. It is inappropriate for complex or creative projects. (Rouse, M. 2013)

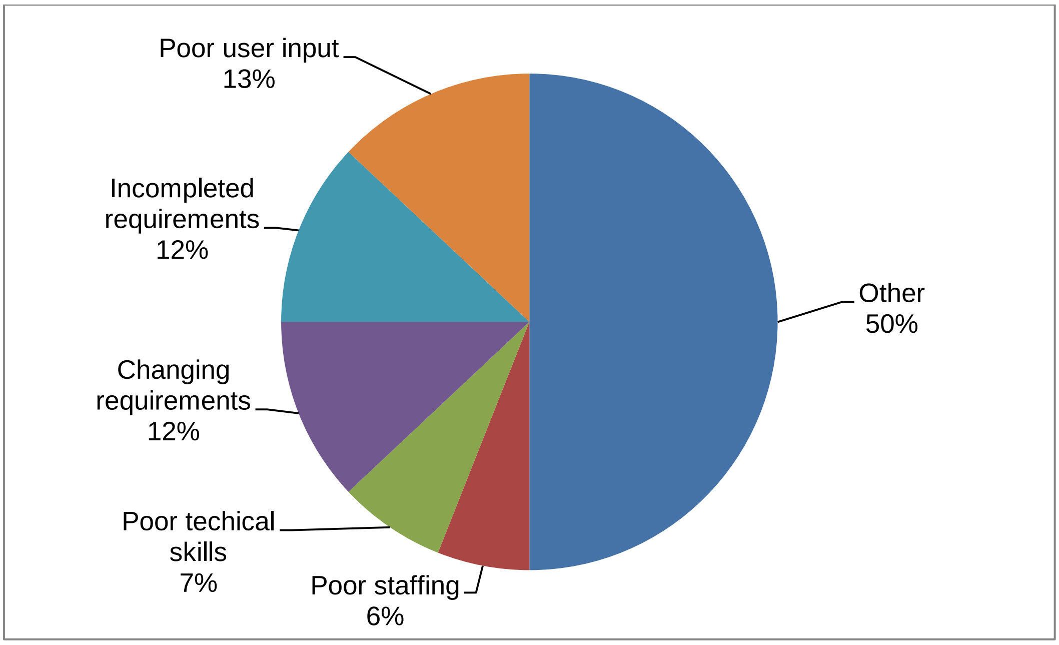


Figure 7; Factors on challenged projects (Johnson, J. et al. 1994).

Johnson J. found in research of over 8000 projects that 37 % of the issues of challenges projects were related to requirements. See the Figure 7; Factors on challenged projects (Johnson, J. et al. 1994).

Meanwhile requirements will change, IID causes more of early change via early development iterations with feedback and practises as multiple requirements workshops. The reality of how people handle the requirements challenge is becoming more iterative. In a research of Chatzoglou, 107 projects were investigated. Only 18 % of the projects tried to complete the requirements in a single early step. 2 % used two cycles of requirements of modification and in 50 % of the projects the requirements analysis was completed over three or more iterations. (Chatzoglou, P. et al., 1996 and Larman, C. 2007)

3.1.1.3 Agile software development model

Software is part of almost all business operations so new software is developed quickly to take advantage of new opportunities and to respond to competitive pressure. Rapid development and delivery is therefore now often the most critical requirement for software systems. In fact, many businesses are willing to trade off software quality and compromise on requirements to achieve faster deployment of the software that they need. (Sommerville, I. 2011)

Agile methodology is an alternative to traditional project management, typically used in software development. Agile software development is a collection of software development methods based on Iterative and Incremental Development (IID). Requirements and solutions change through teamwork between self-organizing, cross-functional teams. (Gartner; Murphy, T.E., 2013)
Agile helps teams to respond to the unpredictability through incremental, iterative work cadences, known as sprints. Agile methods focus on the drive to push quality upstream, or on achieving continuous quality. This includes using for example automated acceptance testing, static analysis, and code reviews, all tied into continuous integration environments. (Gartner; Murphy, T.E., 2013)

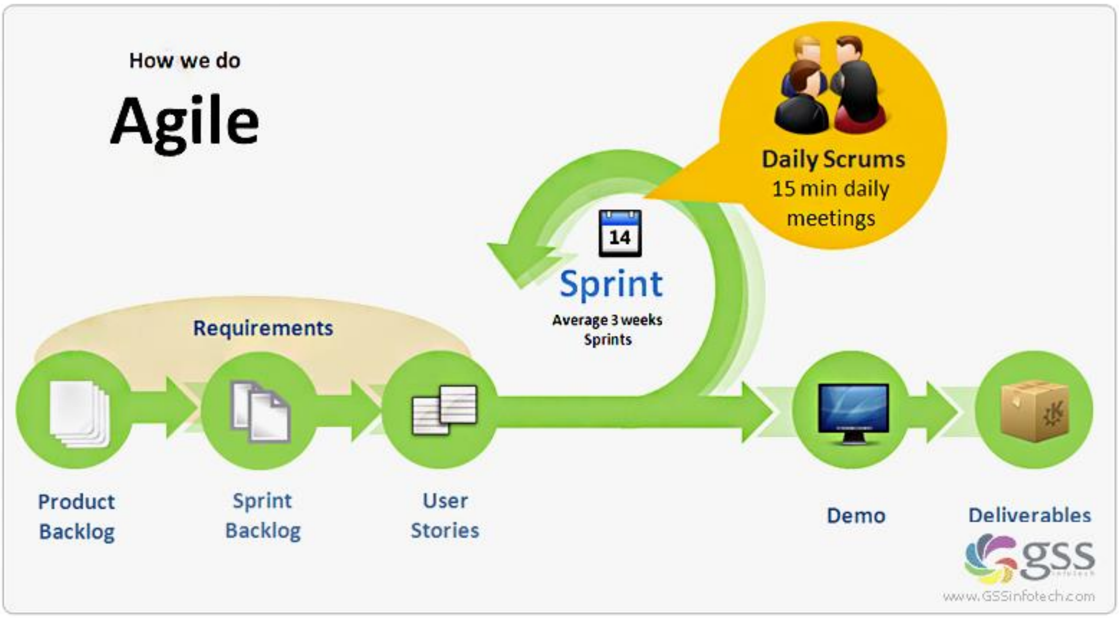


Figure 8; Agile method explained by gss

Agile stimulates adaptive planning, evolutionary development and delivery, a time-boxed iterative approach, and inspires fast and flexible reply to alteration. It is a theoretical structure that encourages forecasted collaboration throughout the development cycle. The Agile Manifesto introduced the term in 2001. Most common Agile methods are Scrum and XP methods. (Larman, C. 2007)
The benefits of Agile methods are;

- High user involvement; short iterations with demos, reviews, requirements refinement, and client-driven iterations are key practices.

- Executive support is endorsed by these practices and particularly through the demonstration of early, genuine results; people like to be associated with projects that that show quick value and progress.
- Clear business objectives for the process. At each project iteration, the project manager should try to find out the most valuable objectives thereby explicating and aligning them with the project.
- Small milestones are at the essence of Agile projects. (Srivatsa, H et al. n.d.)

3.1.1.4 Scrum model

Scrum is an iterative and incremental Agile software development framework for managing software projects and products or application development. Scrum is most often used to manage complex software and product development, using iterative and incremental practices. (Sasankar, A. B. et al. 2011)

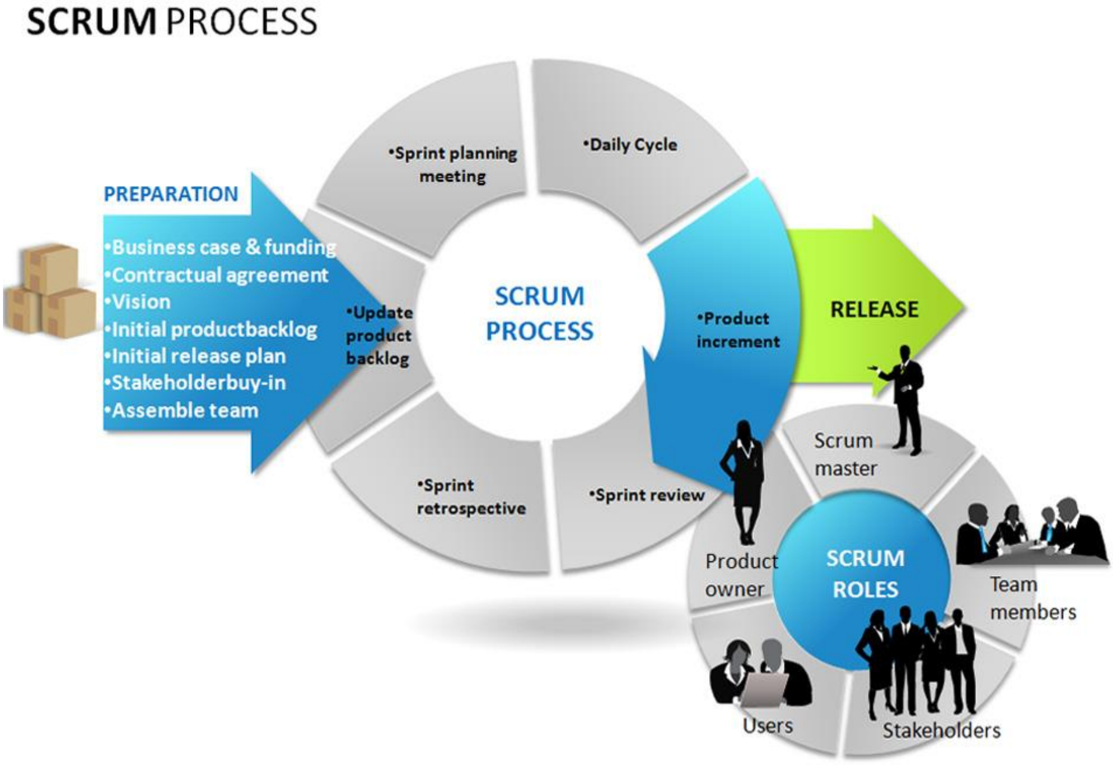


Figure 9; Scrum process by cPrime

Scrum substantially enhances productivity and decrease time compared to classic “waterfall” processes. Scrum processes allow organizations to change smoothly to quickly changing requirements, and produce a product that meets evolving business goals. (Sasankar, A. B. et al. 2011)

A Scrum is a team of eight individuals, ideally. The team acts together and work as tight, integrated units with a single goal in mind. In a similar manner, the Scrum software development process assists a team focus. A Scrum is a methodology for small teams to incrementally build software in complex environments. It is most applicable for projects where requirements cannot be easily defined up. Scrum divides a project into sprints (iterations) for example of 30 days. Functionality is defined before a sprint begins. The goal of the process is to steady requirements during a sprint. (Bhuvaneswari,T. 2013)

A Scrum process benefits the organization by helping it to

- Increase the quality of the deliverables
- Manage better with change (and expect the changes)
- Provide better estimates while spending less time creating them
- Be more in control of the project schedule and state

As a result, Scrum projects achieve higher customer satisfaction rates. (Sasankar, A. B. et al. 2011)

3.1.2 Importance of life-cycle selection

The selection of a software life-cycle model for a project is an important decision. It impacts on project success by affecting the following:

- The software life-cycle’s overall costs
- The distribution of cost over the software life-cycle
- Software development speed
- Software quality
- The ability for tracking and control the project
- The level of risk associated with the project
- Client relationships. (Sasankar, A. B. et al. 2011)

No single software life-cycle model is appropriate for all situations. This is due to the variety in project, system and organisational characteristics. Because of this, a software life-cycle model must to be selected to match each project’s characteristics. (Sasankar, A. B. et al. 2011)

McConnell keeps a record of a number of questions that he pointed should be answered when selecting a software life-cycle model for a software project. He also defines a set of software life-cycle model selection criteria against which the replies to these questions can be compared. A single criterion corresponds to each question.

McConnell’s questions are reproduced below:

- “How well does my customer and I understand the requirements at the beginning of the project?; is our understanding likely to change significantly as we move through the project?”;
- “How well do I understand the system architecture? Am I likely to need to make major architectural changes midway through the project?”;
- “How much reliability do I need?”;
- “How much do I need to plan ahead and design ahead during this project for future versions?”;
- “How much risk does this project entail?”;
- “Am I constrained to a predefined schedule?” (McConnel, S. 1996).

3.1.3 Importance of testing

Testing is necessary because mistakes will appear in the program after coding. Some of those mistakes are not important, but some of them are very expensive or dangerous. It has to test the code because human mistakes will be produced. The Idea is that the developer does not check his/her own code. It is more possible that for example the software tester will spot the errors instead of developer. (ISTQB 2012. ISTQB Certification – Foundation Level syllabus)

The software has to have decent quality. Quality requirements have to match to application such as easy use of graphical user interfaces (GUI) and life threatening functionalities are working for example on airplanes and in health care services. To sustain this quality to a reasonable standard a large amount of effort is required.

Quality assurance (QA) and especially in the development and testing stages the most important part is testing. The later stage of development of program the more difficult it comes to spot and fix errors. Before moving to the next stages, it is important to go through each section during the development that the errors can be found and spotted. In the typical software testing project, the majority of time spent on manual testing is still the largest portion. (Gartner; Murphy, T.E., 2013)

During the development stages unit testing should also be executed. If it is not executed during the development stages, the more it is found bugs and errors during later stages and it becomes more expensive to fix them. Mostly, on the project the main tests are executed during system testing stage. (Glenford J. Myers, 2004)

3.1.4 Why is quality important for a business?

The quality of software is measured with help of testing. Possible methods for measuring are defects found, for both functional and non-functional software requirement and characteristics, for example reliability, usability, efficiency, maintainability and portability. Testing can give confidence in the quality of the software if testing finds few or no defects it can give confidence for the quality. A carefully designed test that passes reduces the overall level risk in the system. If testing finds the most of the defects, the quality of the software system is coming better when those defects are fixed. (ISTQB 2011)

Quality should emphasize three important points:

- Software requirements are the foundation for everything else in the development process.
    - Without a stated goal state, one can never evaluate how closely the finished product complies with the expected product.
    - Verification is the activity of evaluating whether the system is correctly built e.g. that it fulfils its stated requirements.
    - Validation is the activity of evaluating whether the right system has been built e.g. that it fulfils the user’s needs. (Heppenstall, D. 2009)
- Specified standards specify a set of development criteria that guide the way in which software is developed. (Heppenstall, D. 2009)
    - If standards are used then they must be followed.
- There can be a set of implicit requirements that clients/users expect but can rarely articulate during the requirements phase of development.

- If software conforms to its explicit requirements but fails to meet implicit requirements, then it often fails to be successful in the long term and sometimes even in the short term. (Heppenstall, D. 2009)

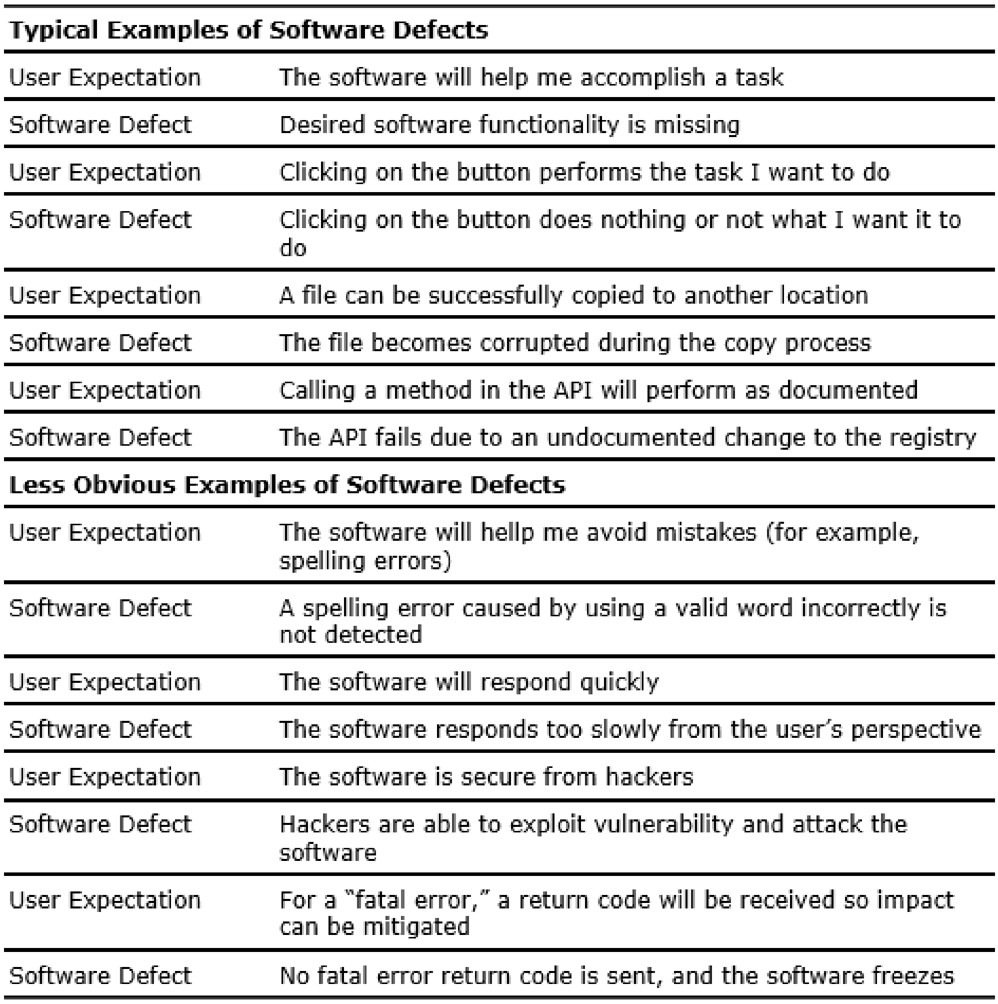
3.1.5 Examples of consequences of serious defects

Defects will affect many ways to software applications. The defect will effect for example on functionality, usability, performance, security, compatibility of software or application. (McDonald, M., et al, 2007)

Software defect is defined by Mark McDonald:

“A software defect is a deficiency in a software product that causes it to perform unexpectedly. From a software user’s perspective, a defect is anything that causes the software not to meet their expectations. In this context, a software user can be either a person or another piece of software. From a software developer’s perspective, a defect is anything that must be corrected in a software work product.”

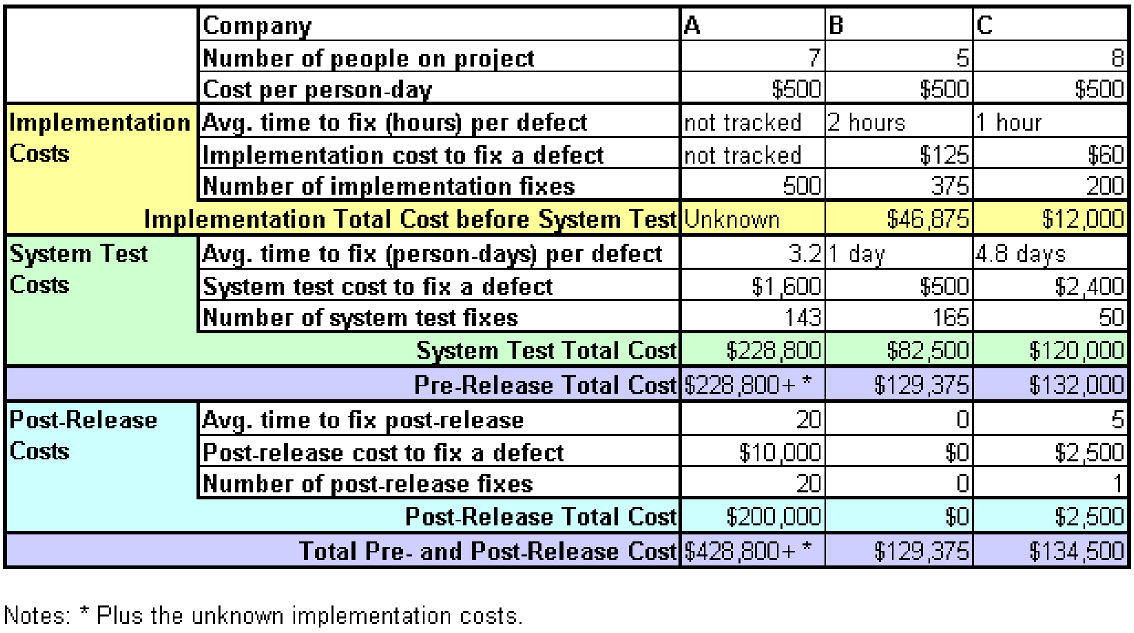
A few examples are listed in Typical examples of software defects (McDonald, M., et al, 2007)



Typical Examples of Software Defects	
User Expectation	The software will help me accomplish a task
Software Defect	Desired software functionality is missing
User Expectation	Clicking on the button performs the task I want to do
Software Defect	Clicking on the button does nothing or not what I want it to do
User Expectation	A file can be successfully copied to another location
Software Defect	The file becomes corrupted during the copy process
User Expectation	Calling a method in the API will perform as documented
Software Defect	The API fails due to an undocumented change to the registry
Less Obvious Examples of Software Defects	
User Expectation	The software will hellp me avoid mistakes (for example, spelling errors)
Software Defect	A spelling error caused by using a valid word incorrectly is not detected
User Expectation	The software will respond quickly
Software Defect	The software responds too slowly from the user’s perspective
User Expectation	The software is secure from hackers
Software Defect	Hackers are able to exploit vulnerability and attack the software
User Expectation	For a “fatal error,” a return code will be received so impact can be mitigated
Software Defect	No fatal error return code is sent, and the software freezes

Table 1; Typical examples of software defects (McDonald, M., et al, 2007)

The most critical for the company are the loss of reputation because of defects and bad quality in software and the costs of fixing the defects. Johanna Rothman has done some example of the costs in the Table 2; Example of costs fixing defects (Rothman, J. 2000).



	Company	A	B	C
	Number of people on project	7	5	8
	Cost per person-day	$500	$500	$500
Implementation Costs	Avg. time to fix (hours) per defect	not tracked	2 hours	1 hour
	Implementation cost to fix a defect	not tracked	$125	$60
	Number of implementation fixes	500	375	200
	Implementation Total Cost before System Test	Unknown	$46,875	$12,000
System Test Costs	Avg. time to fix (person-days) per defect	3.2	1 day	4.8 days
	System test cost to fix a defect	$1,600	$500	$2,400
	Number of system test fixes	143	165	50
	System Test Total Cost	$228,800	$82,500	$120,000
	Pre-Release Total Cost	$228,800+ *	$129,375	$132,000
Post-Release Costs	Avg. time to fix post-release	20	0	5
	Post-release cost to fix a defect	$10,000	$0	$2,500
	Number of post-release fixes	20	0	1
	Post-Release Total Cost	$200,000	$0	$2,500
	Total Pre- and Post-Release Cost	$428,800+ *	$129,375	$134,500

Notes: * Plus the unknown implementation costs.

Table 2; Example of costs fixing defects (Rothman, J. 2000)

In this figure all three companies had different actually costs to fix a defect, because there is no standard cost to fix a defect. For example, Company B had a less complex, smaller product than Company C, so it was expected different costs to fix. The cost to fix a defect changes for different projects and different organizations. (Rothman, J. 2000)

3.1.5.1 The Gmail outage was painful

> "One of the latest software errors that had widely noticed consequences was Google's Gmail outage. The problem in that case was, according to Google, a bug in the software that distributed load between its different data centres.
> The Gmail outage only resulted in people not having access to their email for a few hours. Thankfully no one was killed. Nothing has exploded. It was an inconvenience, and while it was a significant inconvenience to some of Gmail users, it was still just that: an inconvenience". (Pingdom 2009).

3.1.5.2 Undetected hole in the ozone layer

> "The hole in the ozone layer over Antarctica remained undetected for a long period of time because the data analysis software used by NASA in its project to map the ozone layer had been designed to ignore values that deviated greatly from expected measurements.
> The project had been launched in 1978, but it was not until 1985 that the hole was discovered, and not by NASA. NASA did not find the error until they reviewed their data, which indeed showed that there was a big hole in the ozone layer". (Pingdom 2009).

3.1.5.3 November 2000 -- National Cancer Institute, Panama City.

> "In a series of accidents, therapy planning software created by Multidata Systems International, a U.S. firm, miscalculates the proper dosage of radiation for patients undergoing radiation therapy.
> The Multidata's software allows a radiation therapist to draw on a computer screen the placement of metal shields called "blocks" designed to protect healthy

> tissue from the radiation. But the software will only allow technicians to use four shielding blocks, and the Panamanian doctors wish to use five.
> The doctors discover that they can trick the software by drawing all five blocks as a single large block with a hole in the middle. What the doctors don't realize is that the Multidata software gives different answers in this configuration depending on how the whole is drawn: draw it in one direction and the correct dose is calculated, draw in another direction and the software recommends twice the necessary exposure.
> At least eight patients die, while another 20 receive overdoses likely to have substantial health problems. The physicians, who were legally required to double-check the computer's calculations by hand, are indicted for murder". (Garfinkel, S. 2005).

3.1.6 What is a software testing tool?

Software testing tools is a software product that supports one or more testing activities, such as planning and control, specification, setting up starting test databases, test execution and test analysis. A test tool is only a tool when it brings profit; using a tool should not be an aim in itself. Higher productivity and/or efficiency must be achieved when using the test tool. (Koomen, T., Pol, M. 1999)

A testing tool business in the distributed testing tools market is approximately 1,5 billion USA dollars in size and it is going to grow at 6,5 % (compound annual growth rate). There will be seen high growth in the following areas as mobile testing, testing SaaS and PaaS solutions. On the other hand open source tools and tools which are supporting agile methods will manage well, but the challenge will be to measure their impact on the market. (Gartner; Murphy, T.E. 2013).

Program testing and fault detection can be helped significantly by testing tools. They can be utilized for several activities that support testing. For example, test execution tools and test data generation tools are used directly in testing activity.

- Testing tools can aid in managing the whole testing process; for example as tools used to manage tests, requirements, incidents and defects.
- In addition tools are used for reporting and monitoring test execution. The spreadsheet is also seen as testing tool in this context or any tool which is helping testing.
- Testing tool's tasks are to improve the efficiency of test activities e.g. by automation

- To support manual testing activities as test planning, design and reporting. (Ahn, Y., Sampath, V. 2012)

It is ideal to automate tasks which cannot be executed by manual testers for example as large scale performance tests. (Ahn, Y., Sampath, V. 2012)

Automation is a useful option in tasks which requires a large amount of resources in manual testing. Testing reliability can be increased by automating large data comparison or simulating behaviour. (ISTQB 2012, ISTQB Certification – Foundation Level syllabus)

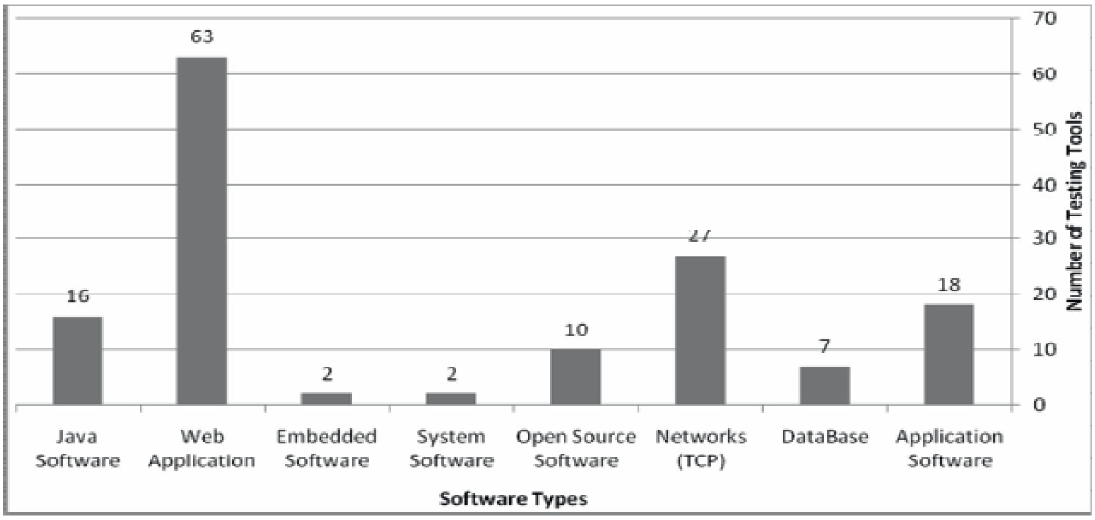


Figure 10; Distribution of the used testing tools over software types (Khaled M. M. et al, 2009).

Kahled M.M. has presented a research of used software tools over software types in 2009. He has found that the most popular are the web applications by 63 used tools, secondly, network (TCP) tools 27 tools and thirds one application software 18 tools. (Khaled M. M. et al. 2009)

Gartner has found in their research that companies have, on average software tools from 3,6 different companies. The drive for a new tool originates from three essential areas;

- Support for new technology
- Support for new processes
- Reducing costs. (Murphy, T.E., 2013)

The greatest satisfaction got companies with support, service and speed of innovation. However the lowest level satisfaction becomes from price, integration and the upgrade experience. There is still a large amount of products lack of success caused by technology support issues or lack of the effectiveness of the solution. Price was the highest variance score recorded in Gartner's research. Many end users felt that they got a higher price or performance of their tools and this is a key value of the product. (Murphy, T.E., 2013)

3.1.7 Benefits of using testing tools

Benefits of using software testing tools are various. As the software industry grows, it becomes strongly competitive and advanced for businesses in order to produce such beneficial quality software. With this competition also comes the reliability of testing tools and deadlines of testing projects which must be met. Systems are more complicated and affect directly human lives. Testing plays a high part in the case of deadlines as testing can take a long time on the software development process. The production of the software and the quality together must be increased for businesses to produce the best possible software. Therefore, testing has to be performed throughout the process of programming the software. Manual testing is often a too slow process. Testing tools and test automation increase the efficiency and assist in meeting the deadlines. (ISTQB 2012, ISTQB Certification – Foundation Level syllabus)

3.1.8 Risks in using software testing tools

One of the greatest risks is perhaps unrealistic expectations. It is important to have clear objectives for what the testing tool can do and that those objectives are realistic. Introducing totally new to the organization is rarely straightforward.

While, there are significant benefits when using testing tools one cannot forget associated risks. At the end, companies may not perhaps reach the original benefits they expected. (ISTQB 2012. STQB Certification – Foundation Level syllabus)

Buying a tool is not a guarantee of achieving benefits. Each type of tool requires effort or investment and time to order to achieve major benefits. There are many risks present when testing tool is introduced and used. (ISTQB 2012. STQB Certification – Foundation Level syllabus)

Following risks have been identified by ISTQB;

- Unrealistic expectations for the testing tools

- Underestimating the time, cost and effort for the initial introduction of a tool
- Underestimating the time and effort needed to achieve significant and continuing benefits of the tool
- Underestimating the effort required to maintain the test assets generated by the tool
- Over reliance on the tool. (ISTQB 2012, STQB Certification – Foundation Level syllabus)

There are also several risks from vendor side;

- The vendor can move out of the business
- Selling the tool to a different vendor
- Retiring the tool itself
- The vendor has poor service. (Simplilearn.com)

In addition to compatibility issues with other tools like requirement management and version control tools can occur. (Simplilearn.com)

3.1.9 What is automated software testing?

Test automation is the process of using a computer program to perform a procedure or user transactions against an IT system. This is typically accomplished by utilizing an automated test tool. Automated testing is normally practised in functional regression testing, performance testing, load testing, network testing and security testing. Automation tools accelerate the test cycle as they can repeat manual testing processes at a significant rate. Some examples of the automated software testing tools;

- HPs Quick Test Professional (QTP)
- Win Runner for Automated functional testing
- Borland’s Silk Test and Compuware’s Test Partner (now owned my Microfocus). (Fewster et al.1999)

Some of the main load testing and performance testing tools;

- HPs LoadRunner,
- Borland’s Silk Performer
- Compuware’s QALoad. (Fewster et al.1999)

Automated software testing is generally the most productive when implemented and executed by skilful resource. (Fewster at al.1999)

3.1.10 Why to automate?

Test automation does not mean automation of the service provided by the software tester. Test automation means using of any a tool that aid testing**.** Automation within the test process can take place in very different ways and generally has one or more of the following aims:

- Fewer hours needed for test execution
- Shorter lead time
- More test depth
- Greater flexibility in testing
- More or faster insight into status of the test process
- Better motivation of the test personnel. (Koomen, T. et al. 1999)

Test automation cannot reproduce the thinking that testers do when they conceive of tests, control tests, modify tests, and observe and evaluate the product. Test automation cannot perform as sapient testing. According to the new doctrine, only manual testing can be sapient*. Automated tests are done by machines and machines are not sapient*. Therefore, automated testing cannot be sapient. At least James Bach has added this new term to testing process; sapience. Sapience is defined in the dictionary as “wisdom; sagacity.” *“*A sapient process is any process that relies on skilled humans.”* (Bach, J. 2012)

Key trends in the automation markets according to Gartner research are;

- Drive for productivity. Organizations are striving to keep up with the quick changes in technology, with new client competences and the recurring need to test on several platforms because of a variety of browsers and devices.
- Changing technology. Concern for the users of test automation tools seems to be the continuously changing and evolving technology.
- Cloud. Network power impacts to testing with tools which control cloud platform, such as Amazon EC2, to create an on-demand load testing.

- Mobile. The mobile market will remodel the background for testing. At the moment, most mobile testing solutions are provided by new companies that only offer device testing support.
- Distributed development. Enterprise software development projects have a habit to be complicated and are often performed in a very distributed approach, whether completely internally sourced or organised in partnership with a System Integrator (SI) or offshore outsourcing provider.
- Agile project management techniques. The evolution of agile development methods continues. It is increasing from developers to include the entire software testing team.
- Constant technology updates. All around the technology industry, contributors have moved to more agile release cycles. Packaged applications have moved from occasional upgrades to regular updates. Browsers and mobile operating systems are updated regularly, and standards are continuing to progress.
- Service-oriented architecture. Testing for services introduces a plenty of complexity and requires organizations to expand minimum standards to operate as well as they at the moment do. Services are expected to provide business-level agility, however companies have conventionally struggled with reuse.
- Open source software. Open-source testing tools remain to make a growth. Open-source tools fit well in companies that are working on smaller projects with a more restricted technological scope, and where the automation is being accomplished by developers.
- Application Life-cycle Management (ALM). According to Gartner, they look at software quality from a collection perspective, including quality management and test execution tools. Leading suppliers have wider solutions including requirements and change management and, in general, offer Application Life-cycle Management solutions. (Gartner; Murphy, T.E. 2013)

3.2 Testing tool classification

There are several different types of testing tools and classifications for them. Each different testing tool is capable of executing different events and each one has different abilities to execute special functions. These are considered to be in the testing tools environment. (ISTQB, 2011)

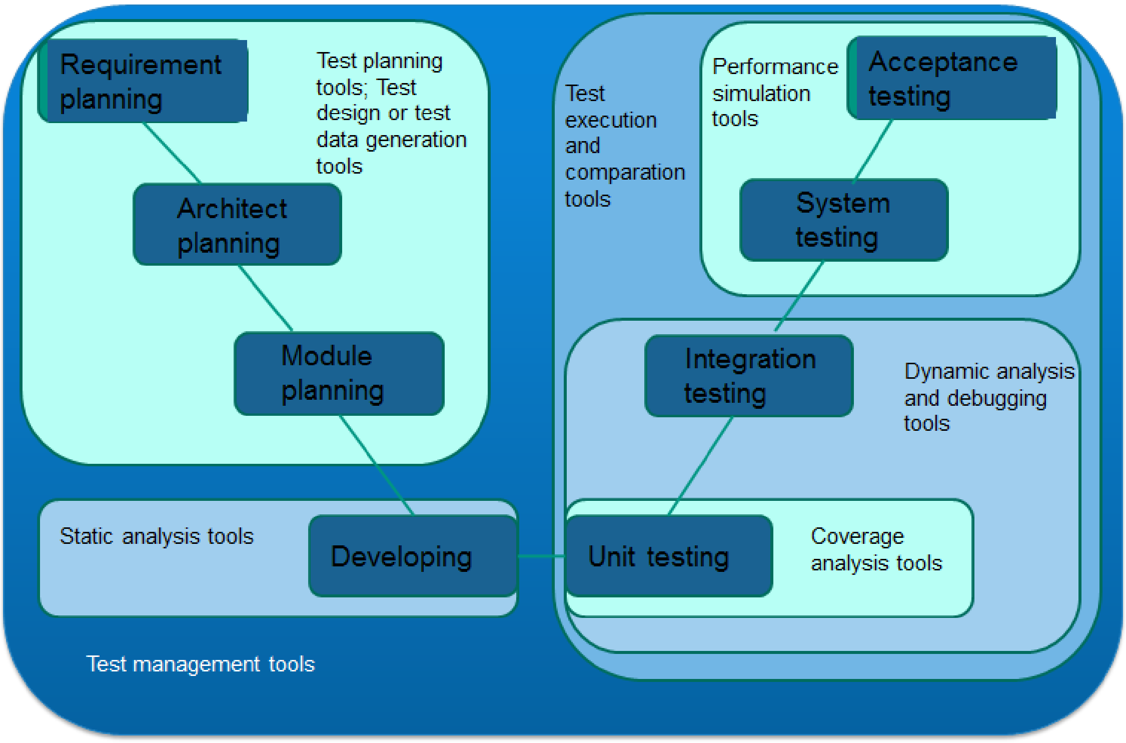


Figure 11; Software testing tools in different development phases. (Fewster et al. 1999)

Over the years there have been much advancement on the different types of testing tools and it will not stop in the future in which they all have their own capabilities of being used for testing. In the following sections there is one of each available and an explanation of what they do. (ISTQB, 2011)

3.2.1 Management tools

Project management is about designing ideas and converting them into a planned, resourced and funded project. The project management process can divide into three stages: Project Initiation, Project Control and Project Closure. (Black, T. 2011)

During the project initiation stage it is necessary to clearly and explicitly define what the project is intended to achieve and define its scope. By defining the scope first, must set a benchmark for the quality of solution at the end of the project. Time and resource allocations are also determined. (Black, T. 2011)

The project control stage consists of monitoring and controlling the progress of the project. It is also about controlling the quality of the product by tracking progress through regular checkpoints and resolving issues that arise during the course of the project. The majority of the work and time spent on a project is during this stage. (Black, T. 2011)

The purpose of the project closure stage consists of two parts:

1) Formally closing the project
2) Passing on any lessons that can be applied to other projects.

There may be some outstanding work that needs to be done and a plan for those things should be done in this stage. There is no need to reinvent the wheel every time you do a project but you do want to become more efficient.

It will be learned how to carry out this type of project better in the future, and will have a better idea of how long the various activities on this type of project will take to do. Any information or documentation from the project should be filed away for future use. (Black, T., 2011)

Following table introduces various management tools.

Test management tools	Test management tools are used to structure automated tests and manual test processes, and to easily manage multiple environments. Quality assurance teams use these types of tools as a single application for managing test cases, environments, automated tests, defects and project tasks. (Koomen, T., Pol, M. 1999)
Requirement management tools	The requirement management tools idea is to store requirements and identify undefined, missing or to be defined requirements. Typical features of the program are traceability of requirements. Requirement management tools are usually interfacing with test Management tools. With these tools you can easily follow up requirements coverage. (Hoffmann, M. et al 2004)

Incident management tools	Definition; “The incident management tool is also known as a defect-tracking tool, a defect-management tool, a bug-tracking tool or a bug-management tool. The name incident management tool describes the tool perhaps best because, incidents may also be perceived problems, anomalies that are not necessarily being defects. However, normally are recorded general information about the failure (not the defect) and the information about the defect that caused that failure”. (ISTQB, 2012)
Configuration management tools	In software engineering, software configuration management (SCM) is the task of tracking and controlling changes in the software. Configuration management practices include revision control and the establishment of baselines. (Koomen, T. et al. 1999)

3.2.2 Execution tools

Execution tools tend to have higher purchasing, training and implementation costs than planning and control tools. But the potential profits in the tool in quality, money, and used time are also greater. Failure risk with these tools is relatively high. Failure in this context is a situation in which the profits will never exceed the costs. (Koomen, T. at al. 1999)

Execution and analysis tools can be supported by several sorts of tools:

- Capture and playback
- Load and stress
- Test coverage
- Test data generator
- Simulators
- Drivers and stubs
- Compilers
- Comparator
- Static analyser

- Query languages
- Debugger
- Monitor (Koomen, T. et al.1999)

The following table is presents additional execution tools:

Test execution tools	Test execution tools are most commonly known category of software testing tools. This type of software testing tools is also known by different names; test running tools, or capture and replay tools. These tools are primarily used for automation of regression testing. They can execute test scripts much faster and more reliable than human beings. Therefore they can reduce test execution time when tests are repeated and/or allow more tests to be executed. (Software testing genius 2012)
The test harness / unit test framework tools	Developers are mostly using these types of tools. These two types of tool are grouped together because they are variants of the type of support needed by developers when testing individual components or units of software. A test harness provides stubs and drivers, which are small programs that interact with the software under test (e.g. for testing middleware and embedded software). Both types of tool enable the developer to test, identify and localize any defects. (ISTQB 2012)
Test comparators	A test comparator helps to automate the comparison between the actual and the expected result produced by the software. 1) Dynamic comparison is where the comparison is done dynamically, e.g. while the test is executed. 2) Post-execution comparison is the other way, where the comparison is performed after the test has finished executing and the software under test is no longer running (ISTQB, 2012).
Coverage measurement	Test coverage is used to measure how thoroughly software is tested. Therefore, developers and suppliers sometimes

tools	use it to indicate their confidence in the readiness of their software. (Yang, Q. et al. 2007) Test coverage tools measure how the test process covers the structure of the software. The measurements can be present at the module level or at the subsystem level. Although structural test coverage of 100%, does not guarantee that testing was complete. (Koomen, T. et al. 1999)
Security tools	Security testing is a process to determine that an information system protects data and maintains functionality as intended. It is performed to check whether there is any information leakage in the sense by encrypting the application or using a wide range of soft-ware's and hardware's and firewall etc. Software security is about making software behave in the presence of a malicious attack. (ISTQB 2012)

3.2.3 Static testing tools

Static testing is a phase of software testing where the software is not actually used. This is in contrast to dynamic testing. Static testing tools are normally used by developers as a part of the development and component testing process. The tool itself is executed, but the source code is in the input data for the tool.

The main goal of static testing is to run a sanity check for the code, algorithm, or document. It is primarily syntax checking of the code and/or manually reviewing the code or document to find errors. This type of testing can be used by the developer who wrote the code, which is in isolation. Code reviews, inspections and walk-throughs are also used. (Van Veenendaal, E. et al. 2008)

The review process support tools	Review process tools are supporting the review process of e.g. documents. Features or characteristics of review process support tools are: • A common reference for the review process or processes to use in different situations.

	• To store and sort review comments. • To communicate comments to the relevant people. • To coordinate online reviews. • To keep the track of comments, including defects found, and providing statistical information about them. • Providing traceability between comments, documents reviewed and related documents; • A repository for rules, procedures and checklists to be used in reviews, as well as entry and exit criteria. • To monitor the review status (passed, passed with corrections, requires re-review). (ISTQB 2012)
Static analysis tools	Static analysis tools analyse the program without running it. The tool does not need any test cases. Also the tool does not recognize how the tool is supposed to behave. The static analysis tool searches for violations against good programming practise and for particular types of programming errors. The negative side of this tool is large numbers of false positives. These tools tend to report a large number of issues that are not bugs. The programmer must manually re-view the list and decide case by case. Sometimes there are too many warnings to sort out. Examples of static analysis tools e.g. FindBugs, JLint, PMD and ESC/Java. (Easterbrook, S. 2012)
Modelling tools	Modelling tools are basically ‘model-based testing tools’ which actually generate test inputs or test cases from stored information about a particular model (e.g. a state diagram), so are classified as test design tools. Modelling tools are generally used by developers and can help in the design of the software. (ISTQB 2012)

3.2.4 Performance testing tools

Performance testing is an umbrella term used for highly transactional types of tests. It is a general definition used to encapsulate Stress, Volume, Load, Soak, Spike and Failover testing to name a few. What follows is an attempt to define, in order of priority, the generalized importance of the different types of performance tests. (Buksh, J. 2011)

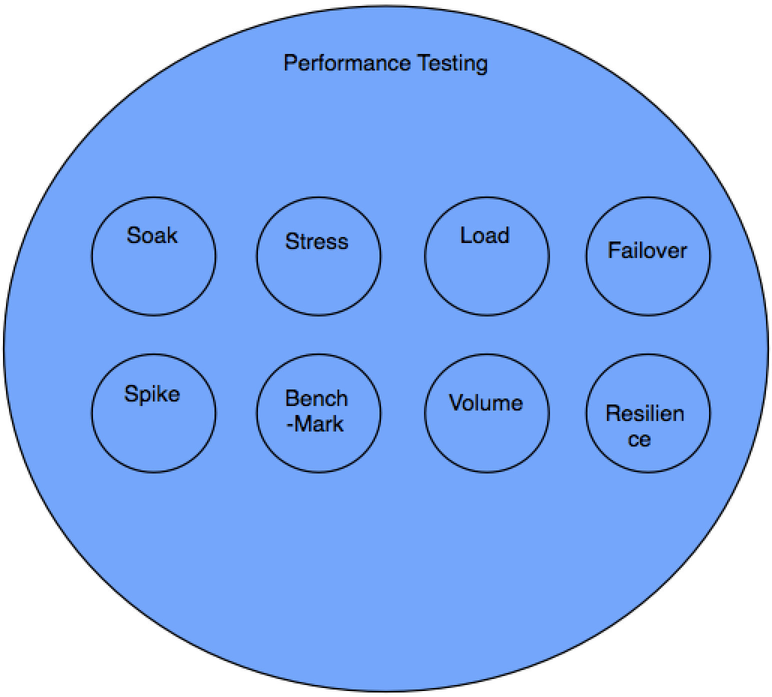


Figure 12; Different types of performance testing. (Buksh, J. 2011)

Margaret Rouse is defining performance testing as “process of determining the speed or effectiveness of a computer, network, software program or device. This process can involve quantitative tests done in a lab, such as measuring the response time or the number of MIPS (millions of instructions per second) at which a system functions. It will be evaluated also qualitative characteristic such as reliability, scalability and interoperability. Performance testing is often done in conjunction with stress testing”. (Rouse, M. 2011)

One of the fundamental advantages of Web applications is that they allow numerous users to entrance the application concurrently. Several users may demand dissimilar services and get permission to changing functionalities all at once. Because, simultaneously user maintenance is vital to the achievement of almost every Web application, we must assess the system competence to accomplish acute tasks during period of normal peak usage. (Nguyen, Q.H. et al. 2003)

The following table presents in more detailed level various performance testing tools.

Load testing tools	Load testing is a kind of performance testing which control a system's functioning under actuality load conditions. This testing helps define how the computer software behaves when several users entry it at the same time. Load testing detects the blockage in the system under various capacities and examines how the system responds when the load is progressively enlarged. (Rouse, M. 2011)
Stress testing tools	Stress testing is the procedure of resolve the competence of a computer, network, program or device to retain an assured level of efficiency under harsh conditions. The procedure can involve measurable tests done in a lab, such as calculating the occurrence of inaccuracy or system crashes. (Rouse, M. 2011)
Dynamic analysis tools	Dynamic analysis is the testing and evaluation of computer software by performing data actually. The objective is to find safety faults in an application while it is running. Dynamic analysis tools are 'dynamic' because they need the code to be in a running state. (ISTQB 2012)
Monitoring tools	To get insight into aspects as memory, CPU, and network usage and time-behaviour, monitoring tool can be used during the test process. They are used to constantly follow of the stage of the procedure in use, in order to have the earliest caution of breakdown, faults or problems and to recover them. Different kind of monitoring tools is for servers, networks, databases, security, performance, website and internet usage, and applications. (Koomen, T. et al. 1999)

3.2.5 Test specs tools

Test design tools are used to help create test cases and test data to be applied for testing. These tools may work from specific requirements document formats, models (e.g., UML), or inputs provided by the test analyst. Test design tools are often designed and built to work with particular formats and particular products such as specific requirements management tools. Test design tools can provide information for the test analyst to use when determining the types of tests that are needed to obtain the targeted level of test coverage, confidence in the system, or product risk mitigation actions. For example, classification tree tools generate (and display) the set of combinations that is needed to reach full coverage based on a selected coverage criterion. This information then can be used by the test analyst to determine the test cases that must be executed. (ISTQB 2012)

Test data preparation tools provide several benefits. Some test data preparation tools are able to analyse a document such as a requirements document or even the source code to determine the data required during testing to achieve a level of coverage. Other test data preparation tools can take a data set from a production system and “scrub” or “anonymise” it to remove any personal information while still maintaining the internal integrity of that data. The scrubbed data can then be used for testing without the risk of a security leak or misuse of personal information. This is particularly important where large volumes of realistic data are required. Other data generation tools can be used to generate test data from given sets of input parameters (i.e., for use in random testing). Some of these will analyse the database structure to determine what inputs will be required from the Test Analyst. (ISTQB 2012)

3.2.6 Specific tools

There are also some specific tools for software testing. The data quality assessment tool is designed to verify the quality of reported data for key indicators at selected sites; and assess the ability of data-management systems to collect manage and report quality data.

Usability testing determines whether an application is useful, findable, accessible, usable, usable, and desirable. Aesthetics and design are important.

How well a product looks usually determines how well it works. Usability Testing identifies usability errors in the system early in the development cycle and can save a product from failure. Usability tool is helping is this process (ISTQB 2012). Examples of usability testing tools; Cage, Grazyegg, Filesquare and Optimizely. (Usertesting.com)

3.3 Methods selecting testing tools traditionally

This section clarifies the methods generally used in selecting testing tools and how testing tools are traditionally evaluated.

It is good to remember that simply purchasing or leasing a testing tool does not guarantee success with that tool and in general in testing. Each type of tool may require additional effort to achieve and lasting benefits. There are potential benefits and opportunities with the use of tools in testing, but there are also risks. Benefits using the tools are when repetitive work is reduced, for example;

- Running regression tests,
- Re-entering the same test data and
- Checking against coding standards.
- Possible to get greater consistency
- Repeatability by executing tests in the same order with the same frequency or derive tests from requirements.
- Can make objective assessments e.g. in static measures and coverage.

When all the information is in the testing tool it is easy access to the information on the test and testing as statistics and graphs about progress, incident rates and performance. (ISTQB foundation syllabus 2011)

Like all tools, testing tools have also risks. Sometimes unrealistic expectations for the tools related to functionality and ease of use may arise. Introduction to new tool can take more time than expected; costs can rise higher than calculated e.g. training costs. Underestimating the time and effort before getting significant advantage of the tool is a common mistake. Sometimes company’s processes need to go through a change because of the tool and more work needs to be done for continuous improvement of the tool. The tools also need continuous effort to maintain and develop test assets generated by the tool. E.g.

test assets need continuous version control. Sometimes people are using the tool wrong way and rely too much on it; for example using automation when manual testing would be more appropriate. When buying new tools it is good to remember check how it is working with other already existing tools in the company. It is more convenient to have testing management requirements, tests, test sets and defects in the same tool. One possible risk is that the company, who has made the tool will possibly make a bankruptcy, sells the business or stopping totally developing any more the tool. Some companies can have poor support for selected tool or they are not efficient making defect fixes and new releases may come very rarely. Open source tools have their own risks too. For example development can be suddenly suspended e.g. if there are new platforms coming in and nobody wants to develop or has ability to develop the tool anymore. (ISTQB foundation syllabus 2011)

3.3.1 Software testing tool evaluation

In these tough economic times managers are pushing to get more and better testing done faster. This same issue was actual already in 1990’s and still is. With automation testing tool’s higher quality and more productive testing is targeted, but setting up and finding suitable tools is not easy to find. (Poston, M.R. et al. 1992)

Are some selection methods better than others? The tool evaluating process itself has not changed since 90’s, Figure 13.

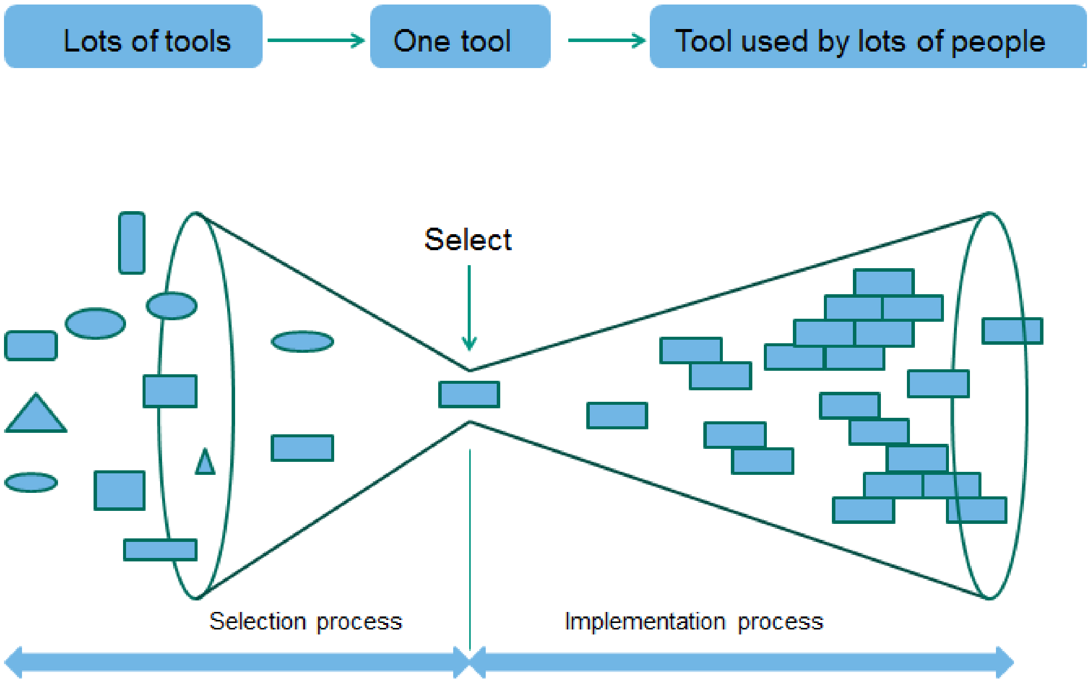


Figure 13; The tool selection and implementation process. (Fewster et al. 1999)

Choosing the applicable software testing tool can be a demanding and frustrating process. The software testing tool should meet the testing organization's long-term and short-term goals. To have a common sense approach and following a few simple procedures to software acquiring and completion methods will lead to a successful implementation of the suitable tool. At the same time a real return on investment (ROI) will be achieved. Testing tools should be brought into an organization to increase the efficiency of a verified testing process - the value of the actual process has already been established within the organization or within the industry. (Johnson, D.W. n.d.).

The first thing an organization must achieve is to record what needs or requirements the testing software is suppose fulfil. For an organization that is new to the acquirements process this can be a rather imminent task. There are three categories or "points-of-view" that must be focused: Management / Organization, Test Architecture and End-User. (Johnson, D.W., n.d.)

The management of the organization or the test team needs to clearly express what the intention is for obtaining testing software. The aim or goal that will be met by purchasing the test software and the expected ROI in terms of person-hours once the tool has been fully implemented. This should be a brief statement on the overall objective, 1 to 3 sentences, not a study or a catalogue of the product competence. (Johnson, D.W. n.d.)

When first approaching the purchase of testing software, test organizations have typically not spend much determination in describing a whole test architecture. Lack of an overall test architecture can direct to outcome a selection that may be applicable in the short-term but ran to further long-term costs or even change of a previously selected tool set. If an architectural structure has been defined then the architectural needs should already be clearly understood and documented - if not then an overall set of architectural rules can be applied. (Johnson, D.W., n.d.)

The end-user needs examination should be detailed study or catalogue of the intended product capabilities as they apply to the testing process. Perhaps a page or more about requirements catalogued in such a way as to accelerate the evaluation process. This is where the explicit and perhaps exceptional product abilities are listed. The most applicable approach is to start from a group of general requirements and then outspread into a list of more specific requirements. (Johnson, D.W., Dev n.d.)

Discovering a list of potential software tool candidates can be investigated by several obvious sources: Generic Web search, Quality Assurance and Testing on-line forums, QA and Testing e-magazines, and co-workers. After a list of potential software candidates has been generated, an estimation of currently available reviews can be done. Important is to to note which products control the largest share of the existing market and which product has the fastest growth rate. This relays to the availability of skilled end-users and their communities. Review the collected information against the needs analysis and create a short list; about 3 to 5 of candidates for evaluation. (Johnson, D.W. n.d.)

Any testing software should be evaluated on-site with a full demo version of the software. When testing any new testing software: set up on a usual end-user system and check following issues:

- Check for .dll- file conflicts
- Check for registry entry issues
- Check for file conflicts

- Ensure the software is operational.

Document any concerns determined during the preliminary set up and search for explanations and the solution from the supplier. (Johnson, D.W., n.d.)

When the selected testing software has been set up to evaluate the software against the earlier requirement analysis. First accomplish any obtainable online lessons and then operate the software to your actuality condition. Document any concerns determined during the evaluation process and look for explanations and the solution from the supplier. Any further requirements found during an evaluation should be documented and utilised to all candidates. (Johnson, D.W. n.d.)

The evaluation process itself will make possible to the evaluation team increase expertise in the product area. It is recommended to make one final effort of all candidates once the preliminary evaluation is finalised. Every software candidate can now be rated against the requirements and a final choice made. (Johnson, D.W. n.d.).

Implementation is clearly not part of the selection process but is a usual breakdown point. Test organizations will often invest in testing software but not in the ability to successfully use it. Some organizations spend a large amount of money in software but not investing funds in on-site training and consulting expertise to prevent a disaster. The software supplier should provide at least a minimum level of training for any sizeable software purchase. The supplier should also supply or recommend on-site consultants / trainers that will guarantee the test organization can take full benefit of the acquired software as soon as possible. (Johnson, D.W., n.d.)

The test organization can avoid much of the interruption by bringing in the right mix of training, consulting, and vendor expertise. By these supporting activities, the supplier can avoid disturbance in any change in the process brings and quickly achieve the advantages that software can provide. (Johnson, D.W. n.d.)

The following five steps as being critical in the test tool selection process according to Testhouse:

- Well defined requirements. The first and most significant step is to identify and document test tool requirements. You have to understand what you are looking for. This is the starting point of the tool evaluation process as well as for the way this tool will be used in the project (Testhouse)
- Tool Shortlist - This will normally consist of tools you or other project members have previously had experience with, plus tools recommended in news groups or testing forums. This list has reduced to few tools which best meet the team’s criteria. (Testhouse)
- Vendors. Customer support services as well as the tool's ability and future plans might make an important influence on your ultimate assessment. (Testhouse)
- Demos. There are trial versions of suppliers’ software available for download. Take the possibility to try out the tool in terms of its features. Although the trial version is often limited and it may not have the full range of features, this is a good way to get a feel of the product. However, this should not be the crucial step in the process as most tools will require a systematic approach to implement the tool. (Testhouse)
- Proof of Concept. The final evaluation will require having detailed knowledge of the subject. Even with hours spent reading supplier’s manuals, handbooks and forums, you may still need to essentially try the tool in the current working environment before acquiring the licenses. It is recommended to arrange the “Proof of Concept” meetings with certified consultants from supplier, to get the deeper overview of the subject. (Testhouse)

3.3.2 Open source tools

The most common choice, historically, is to purchase a tool from a commercial vendor. In some cases, that may be the only viable choice. However, there are other possibilities, such as open-source tools and custom tools, which are also feasible options. This section presents shortly the idea of open source tools. Open-source testing tools continue to make progress, and Selenium in particular has high adoption rate. Open-source tools fit well in companies;

- Working on smaller projects
- With a more limited technology choice
- Where the automation is being performed by developers (Gartner; Murphy, T.E. 2013)

The most significant modification between software created by the open source communities and business software sold by the supplier is that open source software is distributed under authorisation. That guarantees that the source code is accessible to everybody to review, modification, copy, and investigate as they wish. This is the crucial idea of open source: the source code - the language in which the software is written and the key to understanding how the application is running - can be achieved and developed by anyone with the right skills. (Woods, D. 2005)

For almost every stage of testing process there are open-source tools existing. Open source tool does not have a high primary buying cost but there may not have also any official support for the tool and documentation. However, many times open source tools have dedicated following that is willing to fund unofficial support for users. (Gartner; Murphy, T.E. 2013)

Open source tools are also often developed to achieve assured the exact problem or point a single subject; therefore the open source tool may not accomplish all the tasks of a similar supplier tool. The result of this, there should be executed prior to choose open source tools and thorough analysis has to be done. (ISTQB 2012)

The most used open-source tools according to Gartner's research are Geb, Selenium (33% of surveyed users), SoapUI (27 % of users), Sahi, Bugzilla and Jmeter. (Gartner; Murphy, T.E. 2013)

3.3.3 Custom tools

The testing department of the enterprise may find that they have a detailed need for which no supplier or open-source tool is presented. There can be many motives for that: tailored environment, hardware platform or a process that has been modified particular way. In such cases it is probable to think through to create a custom tool. (ISTQB 2012)

The advantages of such a tool are that the tool can encounter the enterprise's need exactly and can run proficiently in the enterprise's environment. Before to publish a custom tool it is crucial to examine the positive and negative issues of the tool: examine the purpose, objective, advantages and possible negative sites. Often custom tools are dependant of the tool maker and therefore the customer tools have to document with care for later maintenance. (ISTQB 2012)

3.3.4 Tools selection process

A question often asked is, which tools are the best? There is not a simple answer As Mark Fewster describes that in his book Experiences of Test automation; “One person’s best car might include space for children and two dogs; another person may prefer speed and performance; another economy.” There is no such a thing as a perfect tool but there are several software testing tools that would be sufficient for a given condition. (Graham, D. et al. 2012)

There is a different kind of testing tools selection processes presented in the literature. One way to select tools is to follow ISTQB recommendations;

- Assess maturity of the organization
- Identify requirements for the software tools
- Evaluate the selected tools
- Evaluate the supplier or service support (open-source tools vs. custom tools)
- Identify internal requirements for training in the use of tools
- Evaluate generally the training needs
- Estimated cost benefits. (ISTQB 2012)

The next phase is to consider the capabilities for each type of tool and list them, for example by questions:

- Is the tool suitable for the thought purpose?
- Can the necessary test data be generated automatically?
- How the needed data selection will be done for the tool (e.g., which test case to execute with which set of data)?
- Will the tool run automatically or will manual intervention be required?
- Does the tool provide adequate logging and reporting?
- Can the design be generated automatically?
- Will the tool run automatically or will manual intervention be required?

- Does the tool provide adequate logging and reporting? (ISTQB 2012)

The example of testing tools selection checklist by Lewis W. E. (2004) is presented in the Appendix 3.

3.3.5 Automation testing tool evaluation

The automation tool is not the most significant characteristic of the test automation. It is needed a tool normally to execute the tests, but the other sides of decent automation are more important than the differences between individual tools in most cases. To have a great automation tool does not guarantee achievement in test automation – the whole test automation system, of which the tool is only one small part, must be designed, customized and sustain. (Ilchenco, A. 2011)

Almost any problem can be resolved by some testing tool, but the complexity and cost of resolutions would be very dissimilar. For example, if your test automation tool has not its own debugger of scripts, the development and debugging of scripts will increase the time required for testing by 30-40%. (Ilchenco, A. 2011)

Success in any test automation effort lies in identifying the right tool for automation. A detailed analysis of various tools must be performed before selecting a tool. This requires a lot of effort and planning. The effort and learning obtained during tool evaluation will help during the execution of the test automation project. (Abraham, C., n.d.)

Several types of Automation test tools can be identified, each suitable for a different testing situation:

- Automated regression testers
- Checklists
- Code coverage analysers
- Control flow analysers
- Interactive debuggers
- Load testers

- Performance and network traffic monitors
- Program complexity analysers
- Simulators
- Spreadsheets
- Synchronized script executors
- Test data generators
- Test management and reporting tools. (Price-Jones, N., n.d.)

A careful selection process is required in order to get the maximum out of the automation test tool. The selection process should have the following stages:
1) Identify business requirements.
2) Estimate the identified business requirements.
3) Accomplish risk/impact analysis of the requirements.
4) Evaluate the available test tools for your particular situation.
5) Select the test tool that best fits your requirements. (Price-Jones, N., n.d.)

3.3.6 Evaluation of vendors

This section presents the leading vendors in software testing markets. Gartner has divided vendors in the four groups; leader, challengers, visionaries and niche players. As seen in the figure below, HP, IBM, Microsoft, Soasta and Borland are seen as leaders in the market. This study will show the same kind of results.

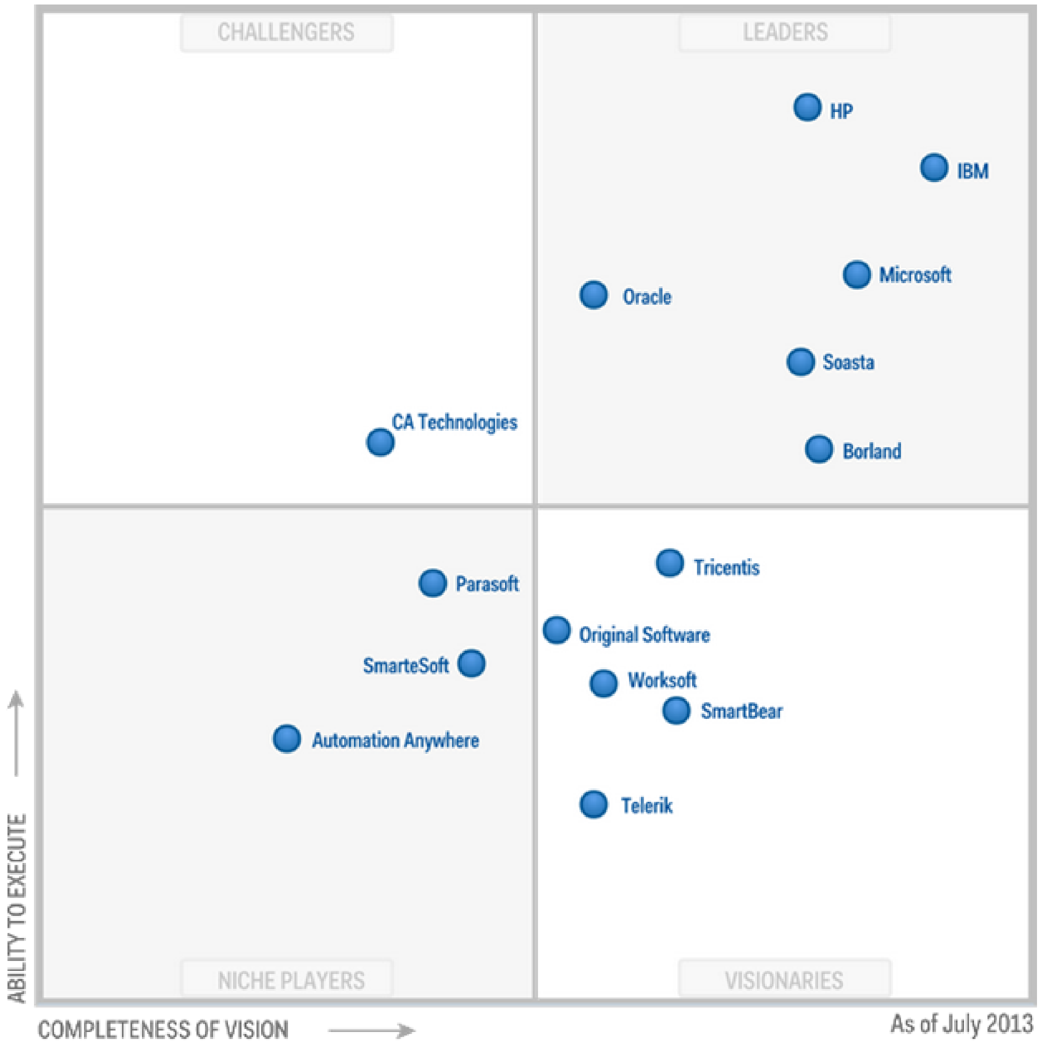


Figure 14; Magic Quadrant for Integrated Software Quality Suites 2013. (Gartner 2013)

There is only one vendor in the challengers category; CA Technologies. In the visionaries category there are for example Telerik and SmartBear. In the niche player category there are Parasoft, SmarteSoft and Automation Anywhere. You can read the whole research in the web-address http://www.gartner.com/technology/reprints.do?id=1-1H9N7L3&ct=130716&st=sb. (Gartner; Murphy, T.E. 2013)

Hewlett Packard (HP) continues to be rated as a Leader in this Gartner’s research. It plays a major role in the software testing market by leading in both market share and annual revenue. However it has seen strong pressure from new market candidates and reviving competition. The company has replied to competition by increasing investment into the product collection and combination. They have also created simplified packaging and licensing methods. For example, Quick Test Professional (QTP) and Service Test have been combined into a single package called HP Unified Functional Testing. (Gartner; Murphy, T.E. 2013)

Strengths

- Market share leader
- Broad technology support
- Wide network of partners, including rivals

Attentions

- Lack of request and support for developer testing
- Premium price with improving, but complex, pricing model. (Gartner; Murphy, T.E. 2013)

Gartner rates IBM as a Leader, with one of the widest collections in the testing tool market:

- Core of test management and test automation
- Static analysis (both code quality and security)
- Unit testing
- And a comprehensive solution for test data through its Optim line. (Gartner; Murphy, T.E. 2013)

IBM has No. 2 position in overall market share and clearly belongs to Leaders group. IBM has increased marketing efforts for this product line with web-based seminars in order to increase customer satisfaction. Users feels that the products provide good value for the money, however noted that integration between the products is unfinished, and there are gaps and lags in the feature collection. (Gartner; Murphy, T.E. 2013)

Strengths

- Platform coverage: Web, legacy, embedded, SOA
- Services and support
- Partners and integrations

Attentions

- Consistency of toolsets due to the acquisition nature of the portfolio
- Reliance on third parties emphasizes Multitool/UI/framework fragmentation. (Gartner; Murphy, T.E. 2013)

Microsoft has become a leader in the software testing market by the implementation of its Visual Studio Team Foundation Server (TFS). It has a compact focus on treating testing as a solid team, including product owners, developers and testers. Microsoft is in third position in the overall market adoption behind HP and IBM, but growing in use more quickly. Microsoft has put effort for a developer-driven quality, which has enlarged over the past few years to create an overall collection. (Gartner; Murphy, T.E. 2013)

Strengths

- Large quantity of training and best-practice materials available through its Microsoft Developer Network
- Integration with the development life-cycle
- Lab Center — test lab management. (Gartner; Murphy, T.E., 2013)

Attentions

- Missing the script-free approach to functional automation
- Slight support for package application testing
- Less valuable for testers outside the Microsoft stack, including those not utilizing Team Foundation Server. (Gartner; Murphy, T.E. 2013)

CA Technologies are seen as a challenger in the software testing market. In the third, visionaries' category there is companies like Telerik and SmartBear. In the niche player category there are Parasoft, SmarteSoft and Automation Anywhere. (Gartner; Murphy, T.E. 2013)

The Gartner research sees that the most important criterion for complete the vision of the company is innovation. Suppliers have one of the three options to complete a vision;

- Create strong innovation
- Target a specific market segment
- Build widespread collection. (Gartner; Murphy, T.E. 2013)

However, Gartner sees also unfulfilled needs in the supplier's product lines. Most common missing requirements are in the areas of unit testing, integration with other areas of the life-cycle, test data management and lab management facilities. Suppliers are using partnerships to fill these gaps, but many leaders in the business are also intensively filling the need. (Gartner; Murphy, T.E. 2013)

3.4 Summary

Testing activities include a wide range of activities before and after test execution. For example, planning and control, choosing test conditions, estimating work load, designing and executing test cases, checking results, evaluating exit criteria, reporting on the testing process and the system under test and finally completing closure activities after the test phase. Testing process includes also reviewing documents and conducting static analysis (ISTQB 2012, ISTQB Certification – Foundation Level syllabus)

The overall main objective of software testing is to check, that the application is working as expected without any errors or bugs (functionality) and that the performance of the application is as expected and meets the needs. (ISTQB 2012, ISTQB Certification – Foundation Level syllabus)

Testing is necessary because mistakes will appear in the program after coding. Some of those mistakes are not important, but some of them are very expensive or dangerous. It has to test the code because human mistakes will be produced. (ISTQB 2012. ISTQB Certification – Foundation Level syllabus)

The quality of software is possible to measure with help of testing. Possible methods for measuring are defects found, for both functional and non-functional software requirement and characteristics, for example reliability, usability, efficiency, maintainability and portability. Testing can give confidence in the quality of the software if testing finds few or no defects it can give confidence for the quality. (ISTQB, 2011)

Quality should emphasize three important points:

- Software requirements are the foundation for everything else in the development process.
- Specified standards specify a set of development criteria that guide the way in which software is developed.

- There can be a set of implicit requirements that clients/users expect but can rarely articulate during the requirements phase of development. (Heppenstall, D. 2009)

Defects will affect many ways to software applications. Defects hinder functionality, usability, performance, security, compatibility of software or application. (McDonald, M. et al, 2007) A few examples of simple defects include;

- Desired functionality is missing,
- Clicking the button does nothing or not what was wanted
- The file becomes corrupted during the copy process.

The most critical for the company are the loss of reputation because of defects and bad quality in software and the costs of fixing the defects. (Rothman, J. 2000)

Software testing tools is a software product that supports one or more testing activities, such as planning and control, specification, setting up starting test databases, test execution and test analysis. (Koomen, T. et al. 1999)

Program testing and fault detection can be helped significantly by testing tools. They can be used for several activities that support testing. For example, test execution tools and test data generation tools are used directly in testing activity. (Ahn, Y. et al. 2012)

- Testing tools can aid in managing the whole testing process; as tools used to manage tests, requirements, incidents and defects only to mention a few.
- In addition tools are used for reporting and monitoring test execution. The spreadsheet is also seen as testing tool in this context or any tool which is helping testing.
- Testing tool's tasks are to improve the efficiency of test activities e.g. by automation or

- To support manual testing activities as test planning, design and reporting. (Ahn, Y. et al. 2012)

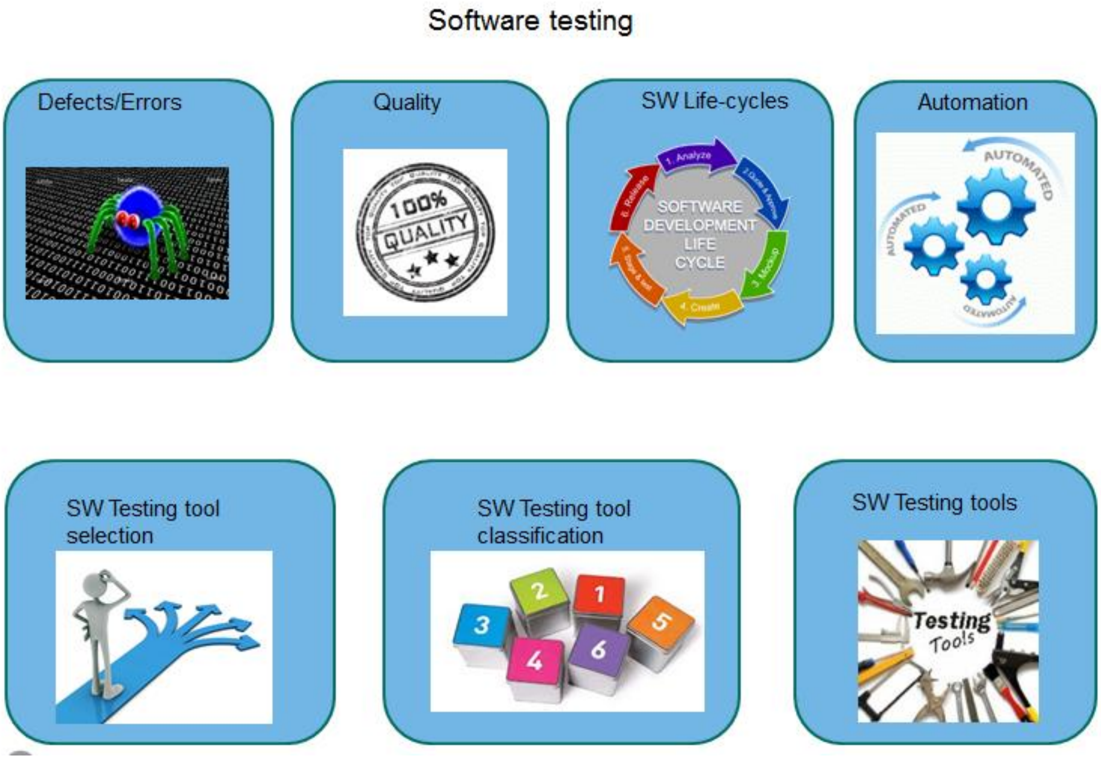


Figure 15; Software testing mind map by author

In the mind map shown above has a few methods affecting to software testing process. These methods have been selected by the author and used as a base for the survey questions and to the study. The study is concerning to the following topics;

- Software testing
- Defects vs. quality
- Testing tool classification
- Software testing life-cycle methods
- Automation
- Testing tool selection process
- Automation tool selection process.

4 Conducting the survey

This section introduces the planning process, content, target groups of the survey and the sending process. In the end, there is presented the analysis of the survey results.

The survey research purpose is to get responses to questions sent to the selected group. There are certain criteria how the group is selected from the whole population. Carefully selected questions are sent to the group. The selected group can be very large as size. When the research is done by a survey - method, the parameters' quantity and content have to plan carefully that the survey is not getting too lengthy. The respondents have to be able to answer to every question unambiguously and without hesitation. (Saaranen-Kauppinen, A. et al. 2006, Tuomi, J. et al. 2009)

4.1 Planning of the survey

This section tells about the process of planning and sending the survey. There were several questions to solve before designing and sending the survey. For example:

- What kind of information included
- How to keep the survey short enough
- Target group selection

4.1.1 Target group of the survey

Core idea was to research software testing tool usage among IT and testing professionals in Finland. It was not found exactly this kind of research material from databases used before and that is why new survey was done.

To narrow down the survey participants, customers from Knowit Oy were selected. It is not possible in survey research to investigate the whole research group (population). The idea was to get the best possible sample of different business areas in Finland. A customer database of over 300 contacts was then investigated and IT & Testing related customer accounts were selected. The list was reviewed by Testing director Kari Kakkonen and Sales director Markus Kääriäinen and selection sample of testing contacts were taken to research.

To gain broad understanding of the business, TestausOSY' was selected as a second source for research contact. In TestausOSY society mailing list everybody, who is interested in testing can join the discussion. From a research point of view this group is arbitrary. The TestausOSY mailing list consists of 1017 people, whose professions and companies are not generally known. Some overlap between these two contact databases occurred. Also especially in TestausOSY mailing list it is possible that there are people who are not exactly or currently belonging to the target research group.

The third and final source for research contacts was Soliditet customer register of Bisnode Finland, also belonging to Knowit Oy. (Soliditet is a Business Area and an auxiliary trade name of Bisnode Finland Oy. Bisnode Group is the leading provider of Credit Information and digital Business Information in Europe.)

4.1.2 Questions of the survey

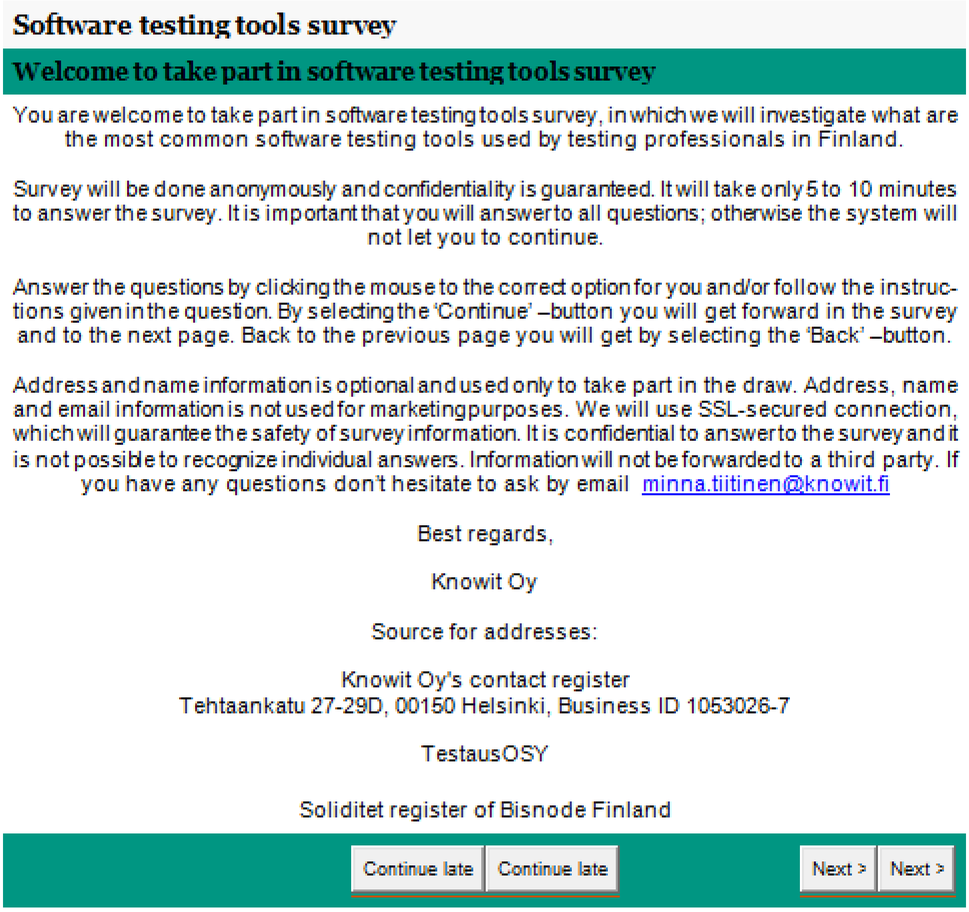
The questions of the survey were planned based on the interest and needs for Knowit Oy and author. Survey questions were divided into two parts; general questions and testing related questions. All together 61 questions were asked; 14 general and 51 testing related questions. The questions were mainly multiple choice questions, but also free comments were possible to give. In the software tool related questions it was possible to select several tools. There was also tool related evaluation of selected tools which was not mandatory. The survey questions are presented in the Appendix 1.

4.1.3 The main content of the survey

The survey structure and related topics were planned in MS Word –document. The survey included the cover sheet, welcome-page, general questions, questions of software testing tools and thank you page. Additionally, there was a reminder cover sheet which was sent after two weeks of the original survey. For TestausOSY a separate email cover sheet and reminder email were planned, because of nature of emailing list. The mailing list was not directly compatible with survey planning application Digium Enterprise.

The cover sheet included the purposes of the survey, information why the survey was done, contact information and source for contacts. The goal was to attract as many people as possible to answer the survey. The cover sheet included a direct link to the web form and to the survey. The draw was organised with the survey and Knowit promised a movie ticket package to attract the people to answer.

First page of the survey was the welcoming –page. It was welcoming the user to the survey and there was contact information if there was anything to ask of the survey later. See the picture below.



Software testing tools survey

Welcome to take part in software testing tools survey

You are welcome to take part in software testing tools survey, in which we will investigate what are the most common software testing tools used by testing professionals in Finland.

Survey will be done anonymously and confidentiality is guaranteed. It will take only 5 to 10 minutes to answer the survey. It is important that you will answer to all questions; otherwise the system will not let you to continue.

Answer the questions by clicking the mouse to the correct option for you and/or follow the instructions given in the question. By selecting the 'Continue' –button you will get forward in the survey and to the next page. Back to the previous page you will get by selecting the 'Back' –button.

Address and name information is optional and used only to take part in the draw. Address, name and email information is not used for marketing purposes. We will use SSL-secured connection, which will guarantee the safety of survey information. It is confidential to answer to the survey and it is not possible to recognize individual answers. Information will not be forwarded to a third party. If you have any questions don't hesitate to ask by email minna.tiitinen@knowit.fi

Best regards,

Knowit Oy

Source for addresses:

Knowit Oy's contact register
Tehtaankatu 27-29D, 00150 Helsinki, Business ID 1053026-7

TestausOSY

Soliditet register of Bisnode Finland

Figure 16; Welcoming -page of the survey

The survey was divided into the two parts; general questions and software testing related questions. General questions were concerning for example the company's size, industry, user's age and profession (Appendix 1).

In the second part of the survey software testing related questions concerning used testing methods, tools and testing tools selection process in the company were asked (Appendix 1). At the end of the survey there was space for volunteer comments generally.

The last page was the thank you –page of the survey and after that it was possible to leave name and email information because of the draw. Last, the survey was sent and saved to the database.

The survey was kept short so that people do not get tired when answering to the questions. Maximum filling time of the survey was between five to ten minutes.

4.1.4 Selected software testing tools

Software testing tools under investigation were selected carefully. Tool selection was done together with colleagues from Knowit and with the instructor Kari Kakkonen. The main criteria in the selection were ISTQB tool classification and commonness of tools. Below is shown the software testing tool classification which was included in the survey. In the survey 127 different software testing tools were listed. The tools are categorized by use:

- Test management tools; 18
- Incident management tools; 6
- Static analysis tools; 21
- Test data preparation tools; 7
- Test execution tools; 19
- Test design tools; 11
- The test harness / unit test framework tools; 17
- Specialized unit test framework, 10
- Dynamic analysis tools; 13
- Performance/ load/ stress testing tools; 13

After the structure and content of the survey was planned and reviewed, the survey was developed into the web form with Digium Enterprise application by Quastback. More information bout the tool on the webpage http://www.questback.com.

4.1.5 Sending the survey

In today’s e-mail rich environment, it is difficult to get good response rates for surveys. In addition some contacts might not be in the direct target group. There was some IT professional who are not working in testing area not at all and do not use testing tools. That is why these contacts were dropped out from the survey.

The survey was sent in two ways; one to the customer contacts through Digium enterprise application and one by email to TestausOSY contacts directly.

- A survey was sent on the 21st of February 2013 and it was given 2 weeks' time to answer to the survey.
- On the 5th of March a reminder was mailed to contacts. This reminder increased replies 40 %.
- The survey was closed 8th of March 2013 and the results were saved to the database.

The total research material was done with 510 contacts. After these conclusions we had the sample of 210 of TestausOSY mailing list, 64 Knowit's customer register and 236 from Solidiet register customer contacts. A survey was sent altogether to 510 people and we got 108 answers and one answer was obsolete. The survey response rate was 21%.

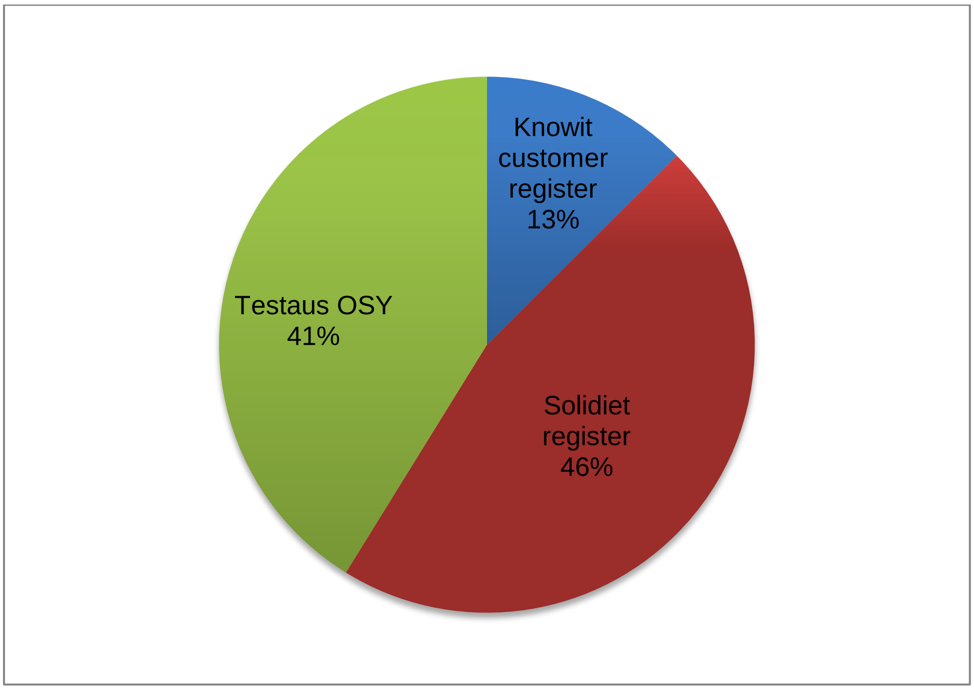


Figure 17; Survey has sent to 510 contacts

Many factors are affecting to response rate as for example connection n to target group, the length of the survey (amount of questions), stimulation to answer (reward), complexity of the survey and the topic of the survey. In a web - survey where the author does not have connection beforehand to respondents, the response rate 20 % to 30 % is seen as very good (Surveymonkey, 2013). The total response rate percentage was 21 %, which in the internet surveys is a reasonable result. It was noticed that TestausOSY contacts were more active than Knowit's and Solidiet registration contacts. The reason for this may be because of above mentioned reasons e.g.; not their profession area, busy times at work, getting too many surveys and emails generally and overlapping of these customer databases.

4.1.6 Reports taken from the database

Once the responses were gathered the reports could be taken into review. It was planned in the beginning that the main reports will be taken from a database by Digium Enterprise. The use of the tool usage was limited and the data were taken to MS Excel spreadsheet and then analysed in pivot tables.

4.2 Analysis of the survey

This section tells about the results of the survey. First general questions analyse was conducted. Then, the results of the software testing tools were analysed. Finally the most popular testing tools according to the survey are presented. The following questions are answered:

- What are software testing tools and why testing is important?
- What are the most popular software testing tools in use?
- Who is using the testing tools?
- What life-cycles models are used in the companies?
- What are the future trends in life-cycle models?
- Do the companies use only one tool or several testing tools?
- How popular are open source tools compared to commercial tools?
- Are users happy to current tools?
- What functionalities are missing from the current tools?
- What requirements there are for testing tools?
- What would be the requirements for new testing tool?
- Recommending how to select testing tools

4.2.1 Analysis of general questions

This section presents the results of general and background questions of the survey. The total response rate was 21 %. Of 107 responses 64 % were men and 36 % women. Most of the responses were from South of Finland (84 %) which was not surprising, because of the nature of the sample. The rest of the respondents were from West of Finland, Åland, East of Finland, Lapland and Oulu province.

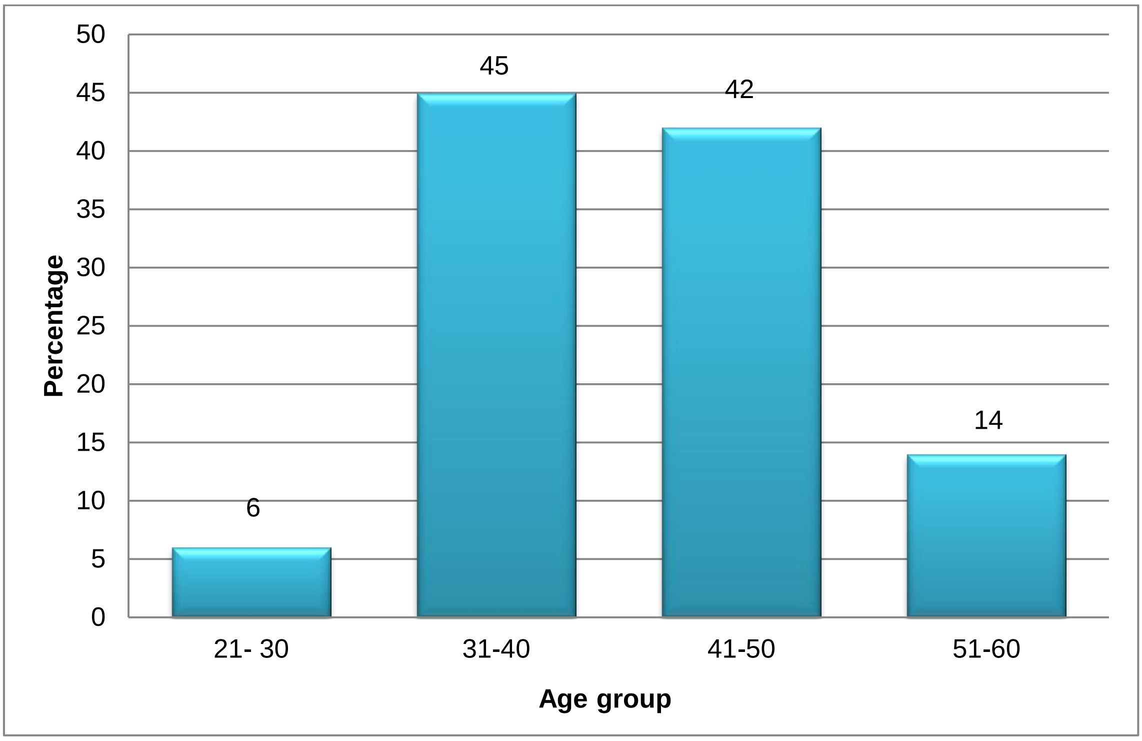


Figure 18; Age of the respondents

87 percentages of respondents belong to age category 31 years to 50 years. From over 60 years of respondents did not receive any responses. Perhaps, because of the nature of IT technology, that it is quite a new industry or usage of IT is lower in that age group.

According to the survey IT-professionals and testing people are highly educated. 42 % have Bachelor or Master Degree and 43% has University degree. In addition 2 % had a doctoral level degree or high school background. Professional school level education has 7 % as seen in Appendix 2. In profession titles there were more dispersion; 37 % has informed profession as Test lead/Test manager/ Test coordinator and secondly bigger is by 28 % some other profession than mentioned in the survey (See Appendix 2).

Testing and IT people seem to work in the large companies; 45 % informed working in the company bigger that 1001 workers. Second came group 101 – 400 workers by 24 %. 1 % did not know how many how many employees there were in the company. Reason can be that people are working as consultants and are not aware of the amount of employees in the company. Also these results can be twisted depending on have the respondents answered by according their employer or company by the customer.

Biggest industries according to the survey were Information and telecommunication 35 % and Finance and insurance business by 21 % as see in the Figure 19. There are no answers at all from Real estate business, Teaching, Warehouse and transfer, Agricultural, Forestry, Fishing business and Hotel and restaurant business in this survey.

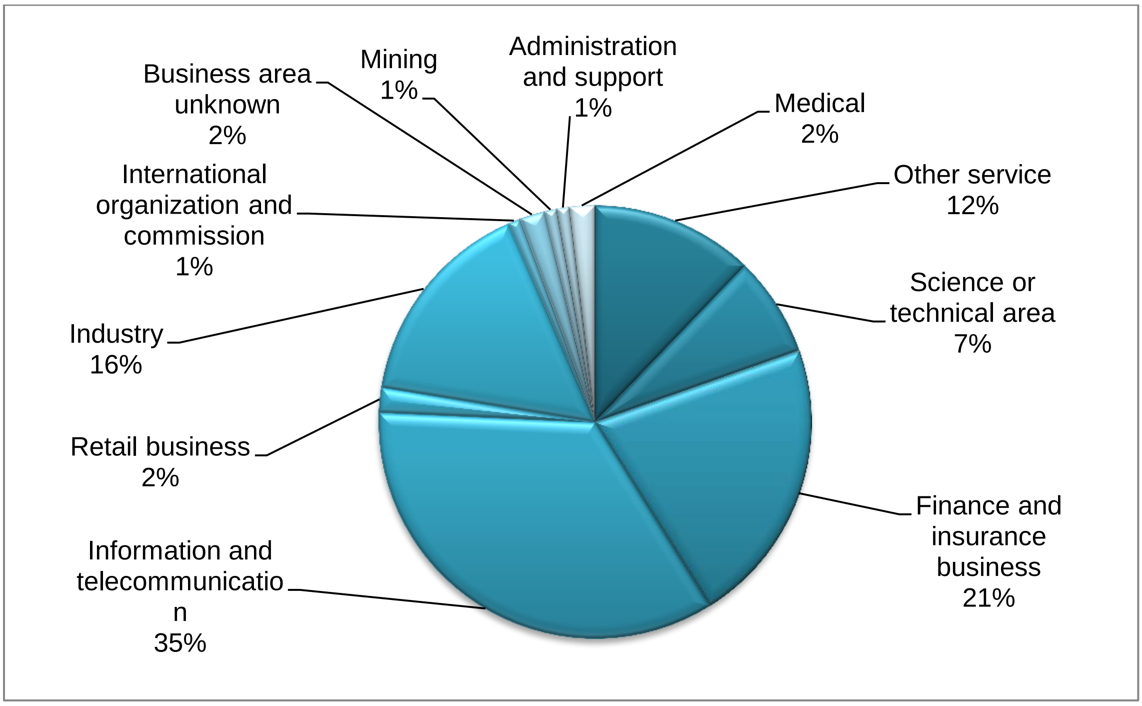


Figure 19; Operating area of the company

51% of the answers indicated that most of answerers informed that they have in the companies use both in-house testers and consultants. 39% have an in-house testing department or software testers and only 3 % informed that they have totally outsourced testing department.

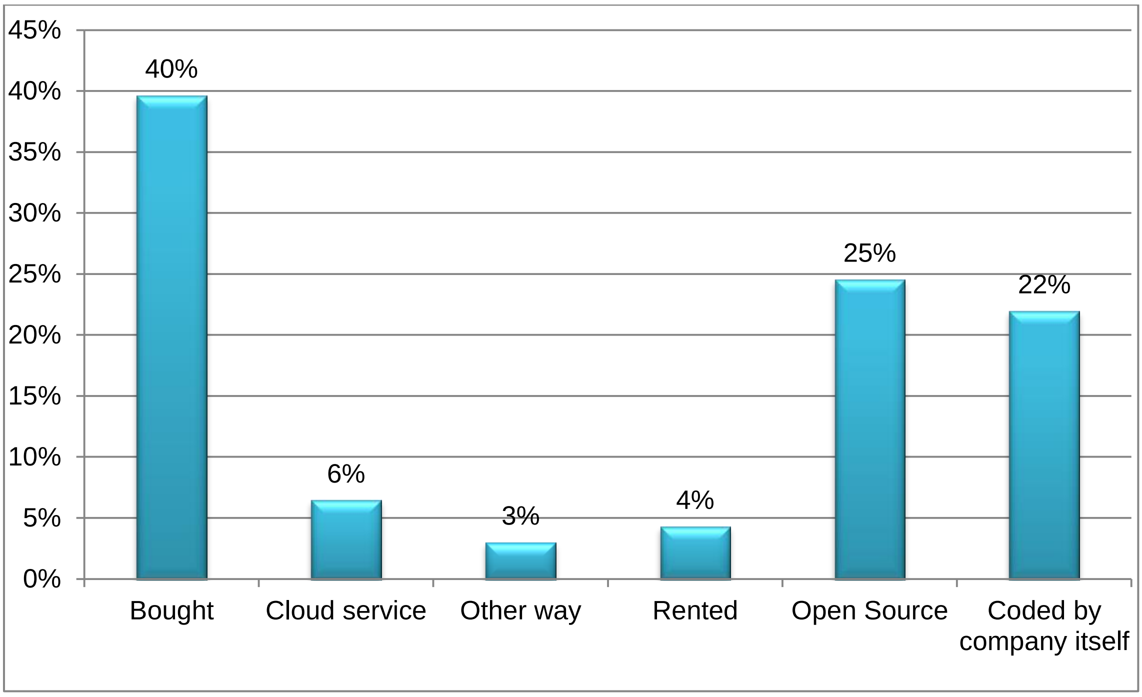


Figure 20; Acquiring method of the software testing tools

It was asked also, how the company has acquired the used testing tools generally. 40 % responded that they have bought the testing tools. 25 % are using open source testing tools and 22 % is using solutions developed in-house. Only 6 % percentage is acquiring testing tools from cloud services, which was quite surprising, as a result. According to this research cloud services are not yet get popular acquiring method (See appendix 2).

There was not found existing researches of testing tool acquiring methods for comparing. But Kahled M.M. (2009) has presented a research of used software tools over software types in 2009. He has found that the most popular are the web applications by 63 used tools, secondly, network (TCP) tools 27 tools and thirds one application software 18 tools. (Khaled M. M. et all 2009) See chapter 3.1.6 What is a software testing tool?

4.2.2 The current and future life-cycle models

The second part of the survey was concerning the usage of the software testing tools and current and future life-cycle models to be used. First of this section of the survey it was asked about the used life-cycle models in the current project.

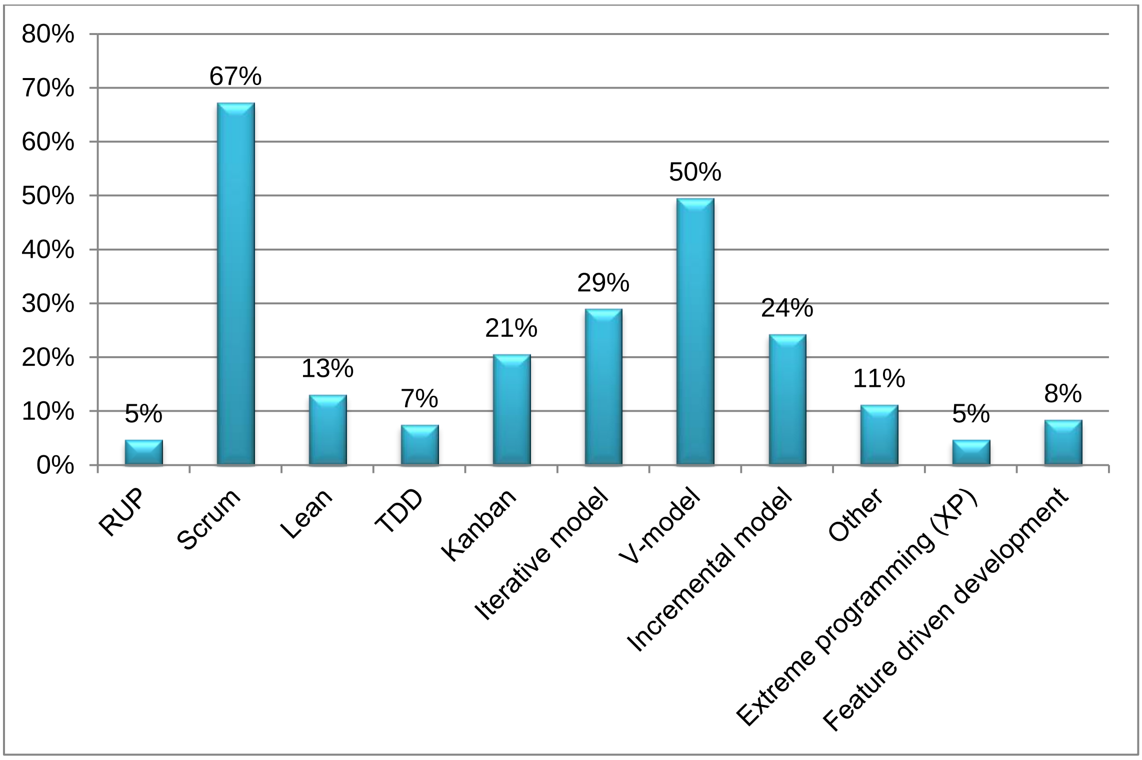


Figure 21; Current life-cycle model in the project

67 % of projects are using Scrum life-cycle but right behind comes the V-model with 50 % and after that Iterative and Kanban are following. Scrum belongs to Agile life-cycle methods. The Gartner has found in their research the same development direction, that Agile life-cycle methods are becoming more popular as seen in the chapter Used life-cycle models in software testing.

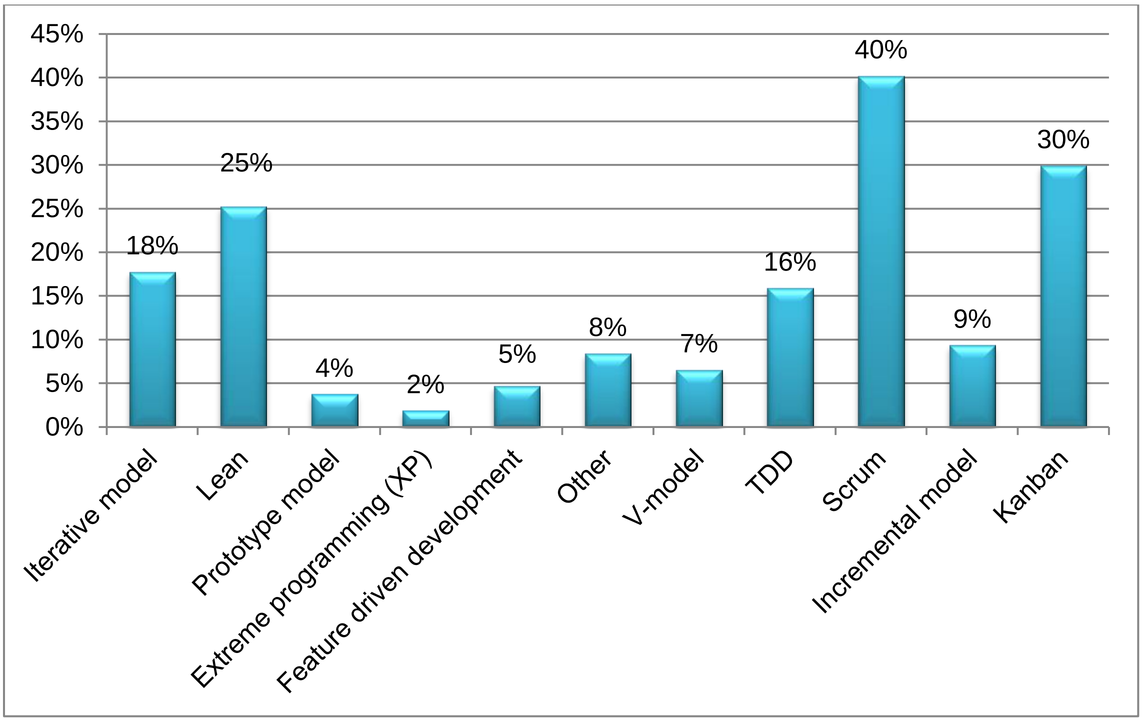


Figure 22; Future life-cycle models

Future life-cycle models were also asked that which one will be the future life-cycle model, Figure 22. Agile methods like Scrum seems to be future trends, Kanban and Lean are following; V-model use is going down.

There is many surveys done of used life-cycle methods. For example, Shine Technologies had 131 international respondents. 85 % were using Agile methods; 58 % Extreme Programming and Scrum 8 %. (Johnson, M. 2003)
Agile Journal's survey was done to 400 international respondents. 80 % were using Agile methods; 28 % Extreme Programming and 20 % Scrum method. (Barnett, L. 2006)
Microsoft survey was done to 492 respondents. Most popular was Agile method 65 %, second Scrum with 65 % and Extreme Programming 5 %. (Begel,A. et. al, 2007)
Also Gartner has mentioned in their research that the evolution of Agile development methods continues. It is increasing from developers to include the entire software testing team. (Gartner; Murphy, T.E. 2013)

Compared to these previous surveys, this study shows that Scrum is the most popular life-cycle method in Finland by 67 % in current projects (Scrum belongs to Agile method). Second most popular is the V – model, which is not at all mentioned in these example surveys. Third one in this study was an Iterative model by 29 % which also is not mentioned in example surveys. See Figure 21; Current life-cycle model in the project

4.2.3 The most popular testing tools according to the survey

This section presents deeper the software testing tools and analyse use of them. First category was Test management tools. Most popular Test Management tools according the survey are HP: QualityCenter with 54 % and the second one is Atlassian Jira by 47%. See Figure 23; Test management tools used among software testers.

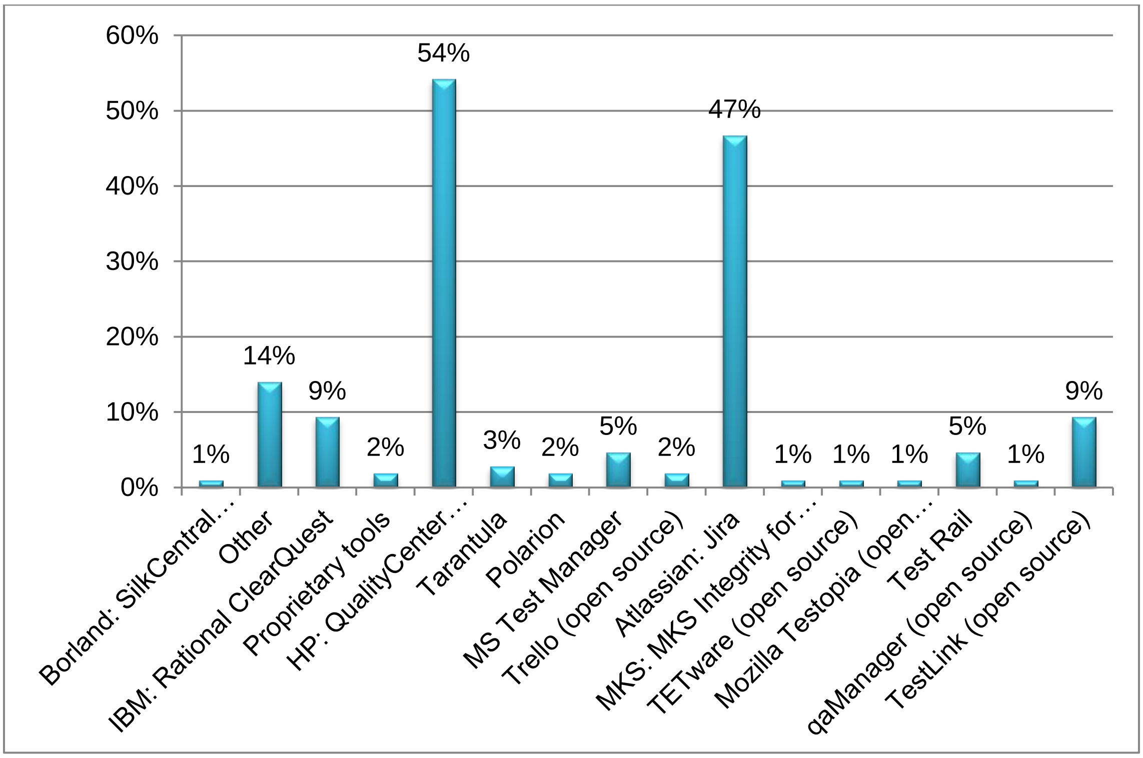


Figure 23; Test management tools used among software testers

Next in the survey it was asked usage about several different testing tools as; Incident management tools, Test preparation tools, Test execution tools, Unit test and test harness, specialised unit test framework tools and Load/Performance/Stress tools. The results are as follows;

Incident management tools;

- Atlassian Jira 41 %
- Open source tools Bugzilla 12 %
- HP; Quality Center 3 %

In test preparation tools;

- MS Visual studio 21 %.
- Second IBM Princeton Softec Optim 5 %

In Test Execution tools;

- Open source Selenium with 45 %
- Other tools 19 %
- HP: Quick test professional 18 %.

Test design tools;

- HP: Quality Center testing tools seem to be most popular very high percentage of 46 %.
- After that is very equal by lower percentages.

Unit test framework and test harness tools;

- Cppunit 27 %
- JUnit 17 %
- .TEST 12 %

Specialised unit test framework tools;

- JBehave by 6 %

Dynamic analysis tools;

- IBM; Rational Purity 5 %

Load/Performance/Stress testing tools;

- Open source tool Jmeter with 24 %
- HP – Performance Center 22%

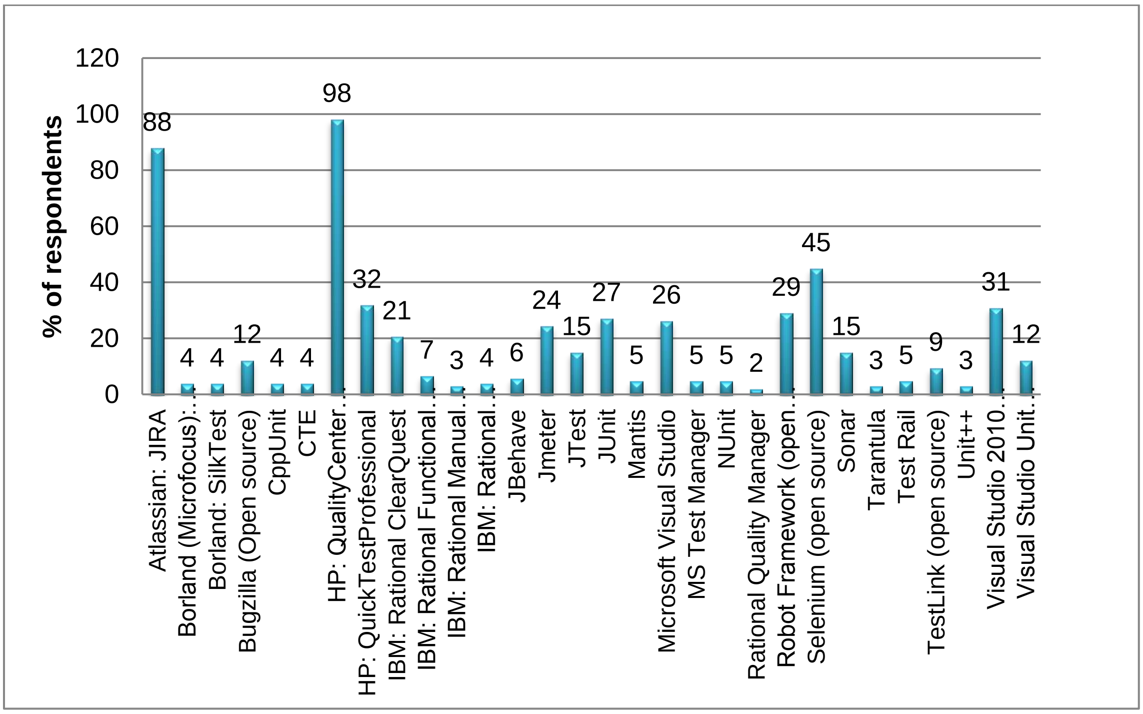


Figure 24; Overall usage of software tools

In overall analysing of all software testing tools the HP's Quality Center is the most popular tool by 98 % according to the survey. For second came Atlassian JIRA by 88 % and third one open source tool Selenium by 45 %. These tools are used in different testing phases. See much more information of different testing tools classifications presented in Appendix 2.

The Gartner has concentrated in their research to Automated Software Quality (ASQ) assurance market which is a subsegment of the overall software development life-cycle market. It consists of three areas;

- Test management
- Automated stress and load testing
- Automated functional and regression testing

The Gartner has not set up vendors in order or presented percentages, but has divided vendors in the four groups; leader, challengers, visionaries and niche players. According the Gartner's research leaders were HP, IBM and Microsoft. The very same results have founded also in this study about vendor's popularity in Finland.

The most used open-source tools according to Gartner's research are Geb, Selenium, Sahi, Watir, Bugzilla and Jmeter. (Gartner; Murphy, T.E. 2013) See Chapter 3.3.1 Software testing tool evaluation. In this study most used open source tool were Selenium and the second Robot Framework. The third was Bugzilla.

Testing tools are used by testing professionals and developers according to the survey. Most used tools in the company were commercial testing tools and open source tools (See Appendix 2).

4.2.4 Purpose of testing tools

Testing tools are utilized for several activities and several tools are in use same time. Typical testing activities such as planning test cases, reporting, defect control and management, and test execution have the lead in this category. See Figure 25; Purpose of the software testing tools.

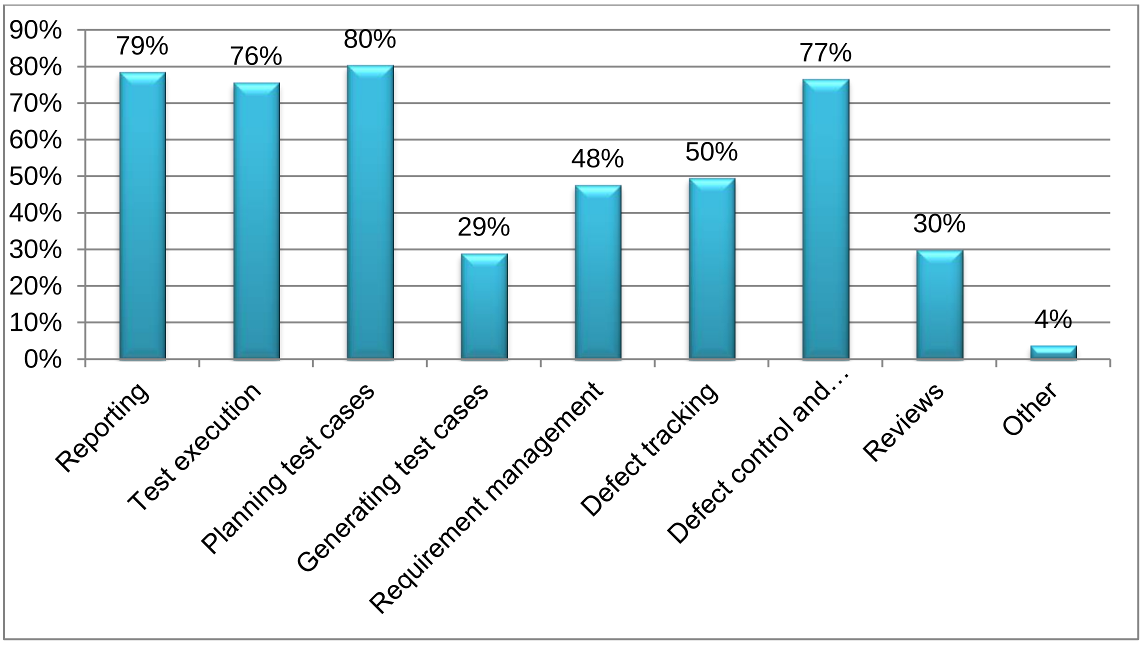


Figure 25; Purpose of the software testing tools

All these activities are quite equal. Planning the test cases got 80 % of all the answers. Secondly reporting by 79 %, defect management 77% and test execution 76 %. Other asked question in this category was defect tracking, requirement management, generating test cases and reviews. See Appendix 2 for more information. There was not found previous researches for comparison.

4.2.5 Testing tools acquiring methods

Normally software testing tools are acquired by buying. Next used tools are open source tools and proprietary tools. According to this study cloud services are not yet a popular acquiring method as seen in the Figure 26; The acquiring methods of software testing tools.

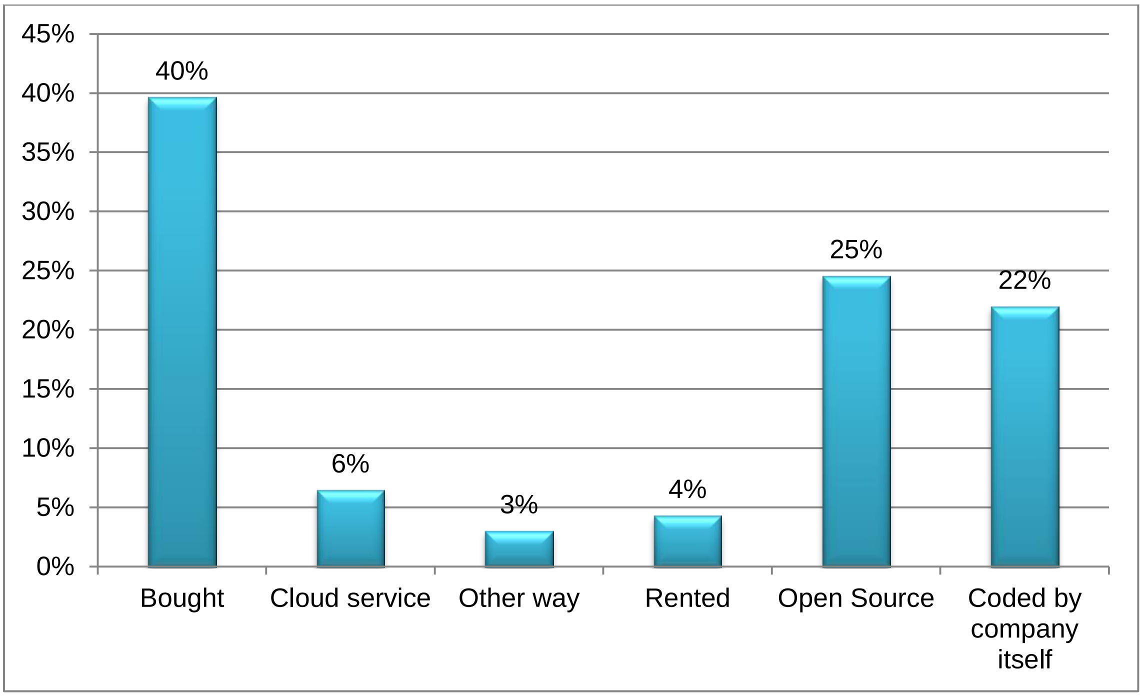


Figure 26; The acquiring methods of software testing tools

Only 4 % of software testing tools are rented according this study.

When asked who are selecting the software testing tools the answer was that Test managers/ Test leads/ Test coordinators. After that tools are selected by software testers and thirdly by management of the company. This same topic was also analysed cross checking the answers which one of the respondents have had a possibility to participate to tool selection process versus role they have in the company. Results seen in the Figure 27; Participation in the tool selection process vs. role in the company.

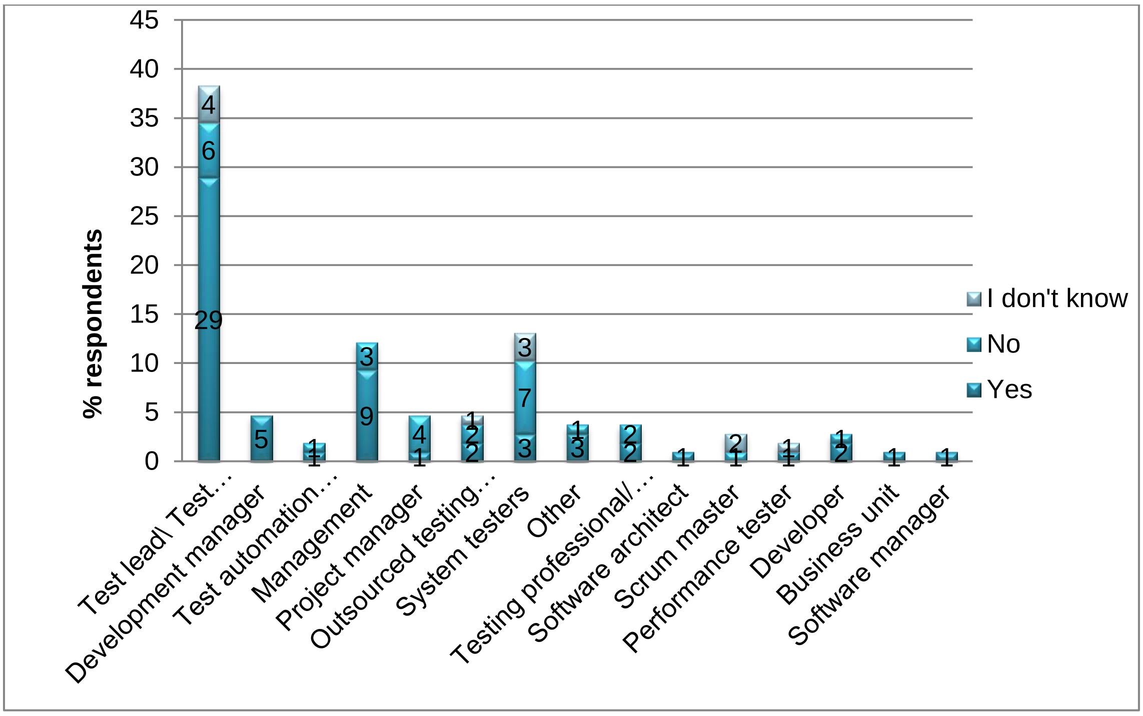


Figure 27; Participation in the tool selection process vs. role in the company

This gives the same result that Test managers/ Test leads have had the possibility to participate in the selection process by 29 % and Management by 9 %. Software testers are very rarely involved in software testing tools selection process, which is a pity, because the exact end users of the tool should be also participating in the selection process. There was not found previous researches for comparison.

4.2.6 Do the companies use office tools for testing?

This section analyses the companies use of several testing tools at the same time. Secondly office tools are analysed are they still in the use. In this context it is meant by office tools the spreadsheets, calculation tools and word processing tools.

According the survey results, it was clear to see that companies are using several testing tools at the same time. In the survey it was possible to select several tools.

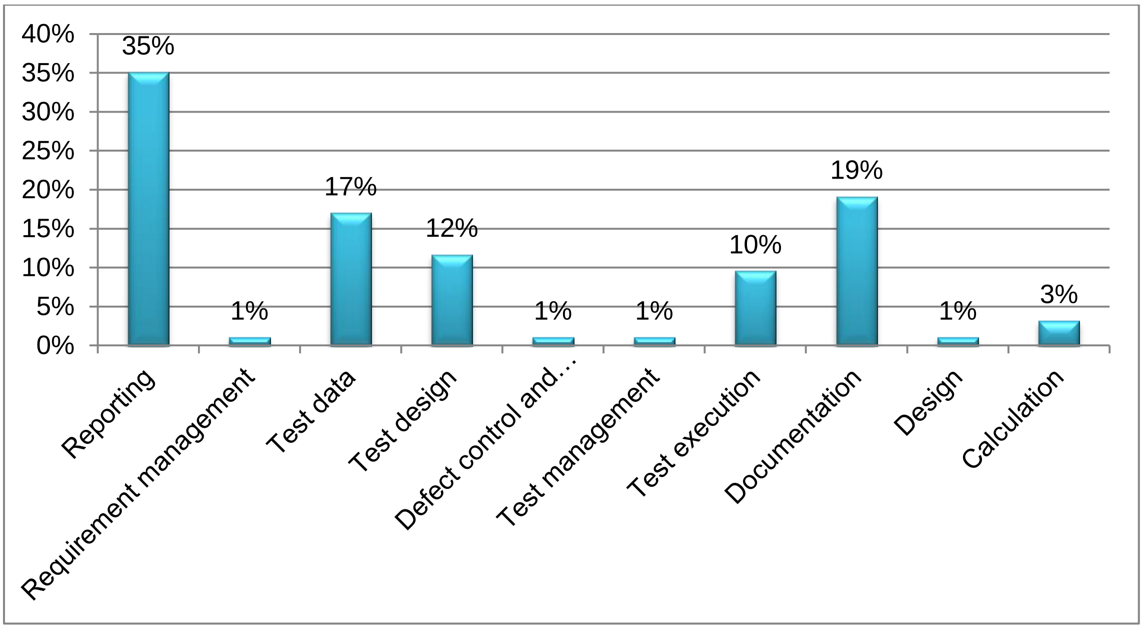


Figure 28; So called “office tools” are used same functions as testing tools

It was interesting to see that there are some companies still using word processing and spreadsheet tools. 83 % of respondent said that they are using for example spreadsheet and word processing program helping the testing. These tools are used for the same functions as the software testing tools. There was not found previous researches for comparison.

4.2.7 Popularity of open source tools vs. commercial tools?

One interesting question was that how popular are the open source tools compared to commercial tools. This was asked two ways;

- What kind of testing tools are in use?
- Which way the testing tools are acquired?

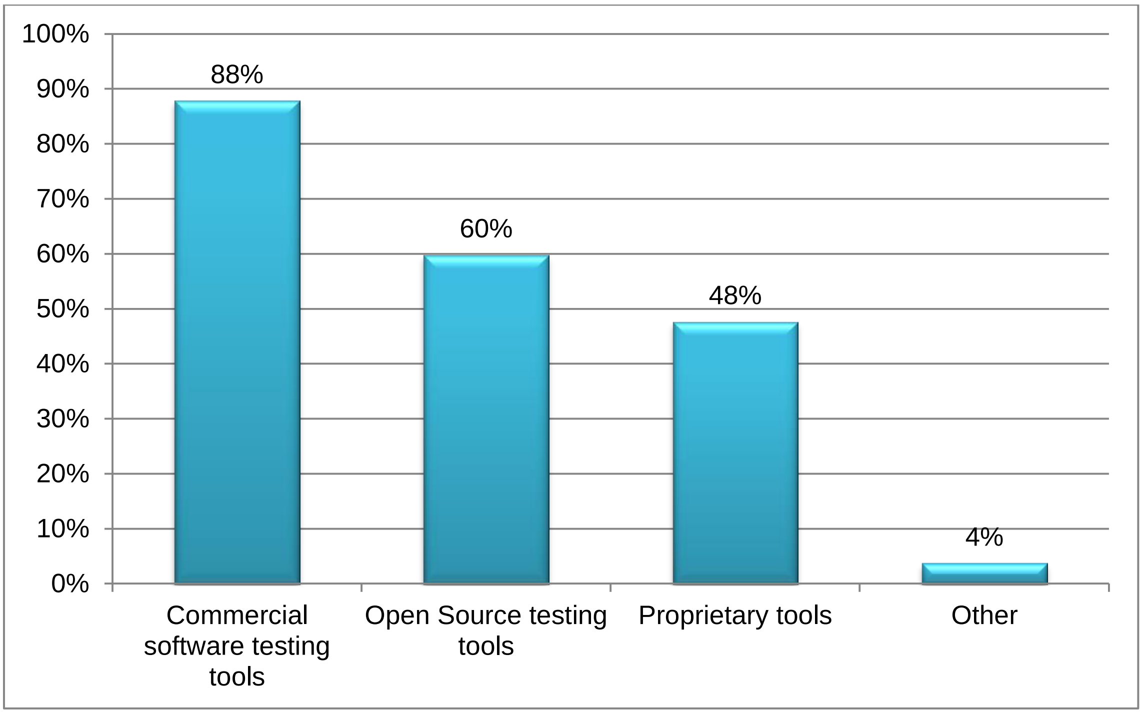


Figure 29; Used software testing tools

The survey showed that many tools are used side by side. The companies are using commercial software testing tools according the survey by 88 %. Open source testing tools were secondly popular by 60 %. Third one was the tools coded by the company itself.

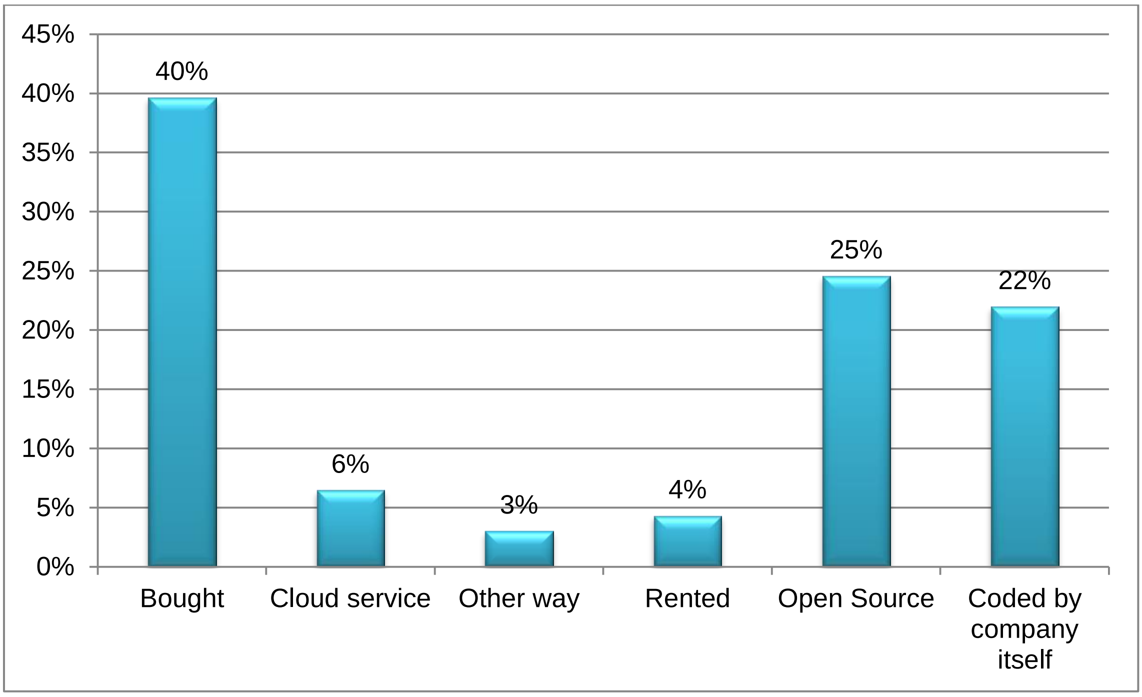


Figure 30; Acquiring method of the software testing tools

The second question was concerning of the ways acquiring software tools. Typically software testing tools are bought. Next acquisition methods are open source tools and proprietary tools coded by company itself. Surprisingly, the survey brought up that the cloud services are not yet a popular acquisition method, only 6 %. There was not found previous researches for comparison.

4.2.8 What would be the requirements for a new testing tool?

Testing tool selection for a company is a complicated process. The survey solved which functionalities are affecting when acquiring a software testing tool.

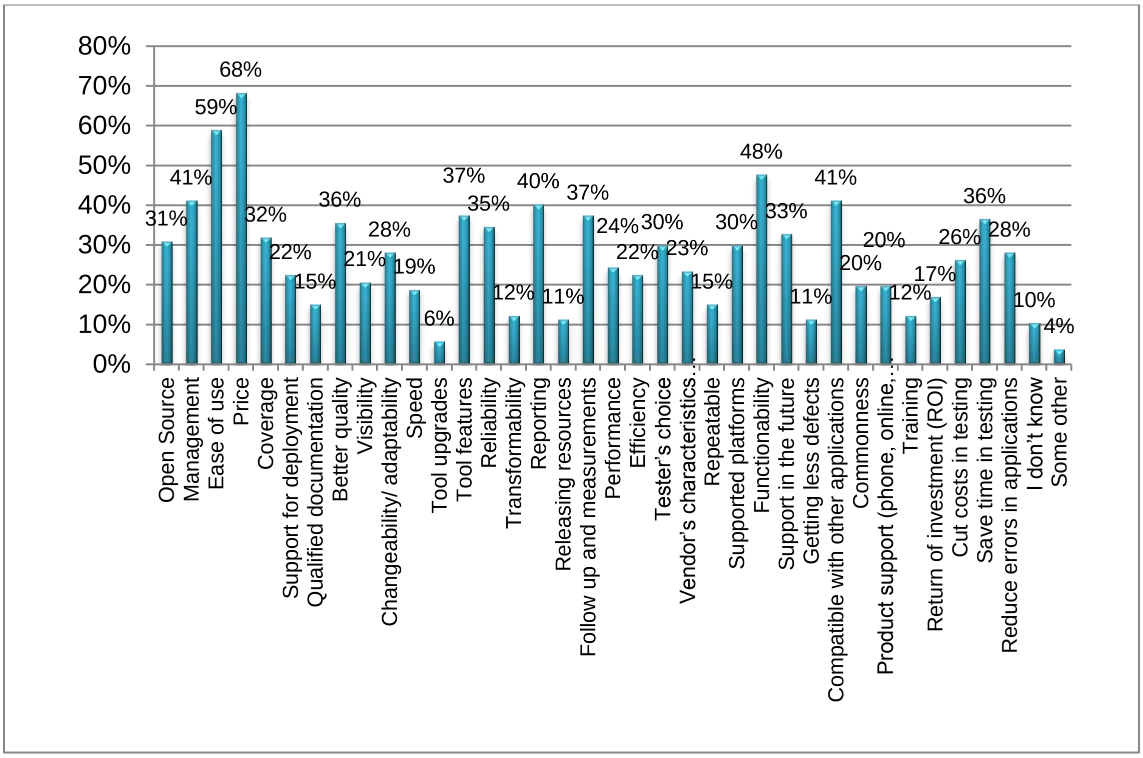


Figure 31; Functionalities affecting to testing tool selection process

In total 35 different functionalities affecting testing tool selection were listed. The results were divided quite equally but two main things popped out in the end; 68 % think that the prices are the most important thing and secondly ease of use by 59 %. Third one was functionality. The result is quite interesting because there are two totally different kinds of answers; the price is something the user can not affect and on the other hand generally the management is selecting the tools based on the price. The second functionality ease of use is something that the user of the tool appreciates. The management usually does not have experience of this.

Important qualities or functionalities for respondent were;

- Functionality
- Compatible with other application

- Management of the tools and
- Reporting were important qualities.

Compared to ISO/IEC JTC 1/SC 7 N 5401 standard the quality characteristics that the test target must satisfy are classified as functional requirements or non-functional requirements as described below.

- Functionality, suitability, reliability, usability, efficiency, maintainability, portability, compatibility and security. (ISO/IEC 25010: 2011)

On the other hand Illes has mentioned in his article important criteria for software testing tools as following;

- Functionality, reliability, usability, efficiency, maintainability, portability, general vendor qualifications, vendor support, licensing and pricing. (Illes, T. et al. 2005)

Thesis results are quite similar compared to standard (ISO/IEC 25010: 2011) and Illes's criteria.

4.2.9 Training of software testing professionals

In software testing it is important to keep yourself up to date of testing trends. Software testing has progressed quite rapidly from the 1950's to nowadays professional testing methods. Training is one method to catch up with new information.

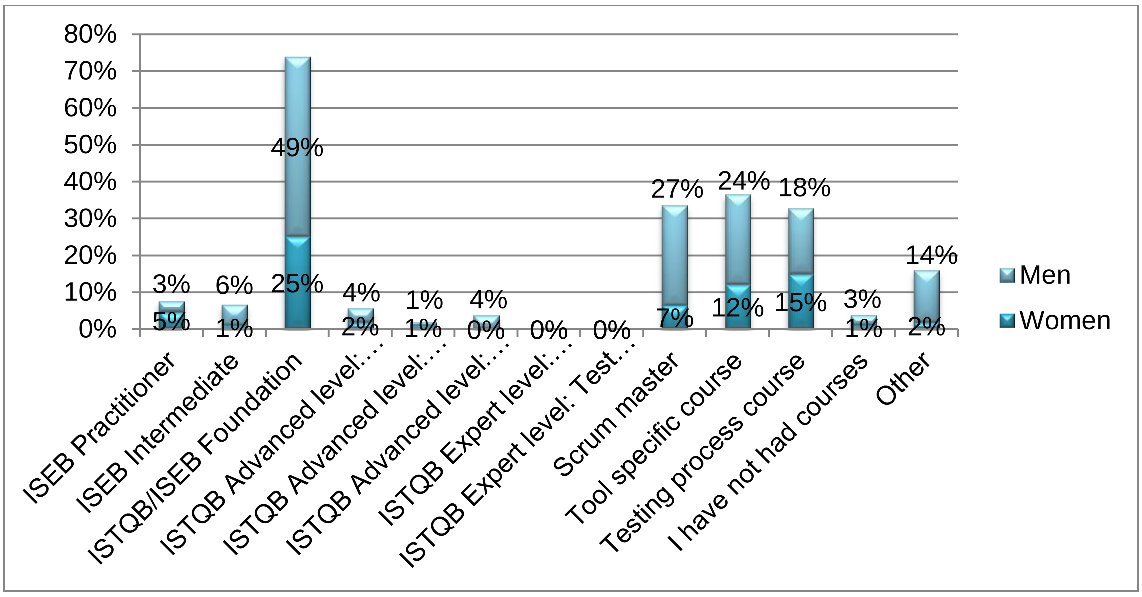


Figure 32; Respondents' training

The question of testing specified training was asked in the survey and 74 % of respondents have an ISTQB/ISEB Foundation course. The total amount of respondent 64 % were men and 49 % of them had ISTQB/ISEB Foundation course. On the other hand, 36 % of respondents were woman and 25 % of women have an ISTQB/ISEB Foundation course. Next popular training in the survey was tool specific courses and third one scrum master course. That is followed Testing process courses and ISEB intermediate courses. There was not found previous researches for comparison.

4.2.10 Evaluation of most popular testing tools and vendors

In the survey, satisfaction of the used software tools and vendors was asked. Satisfaction among test management tools seems to follow the order of tool popularity.

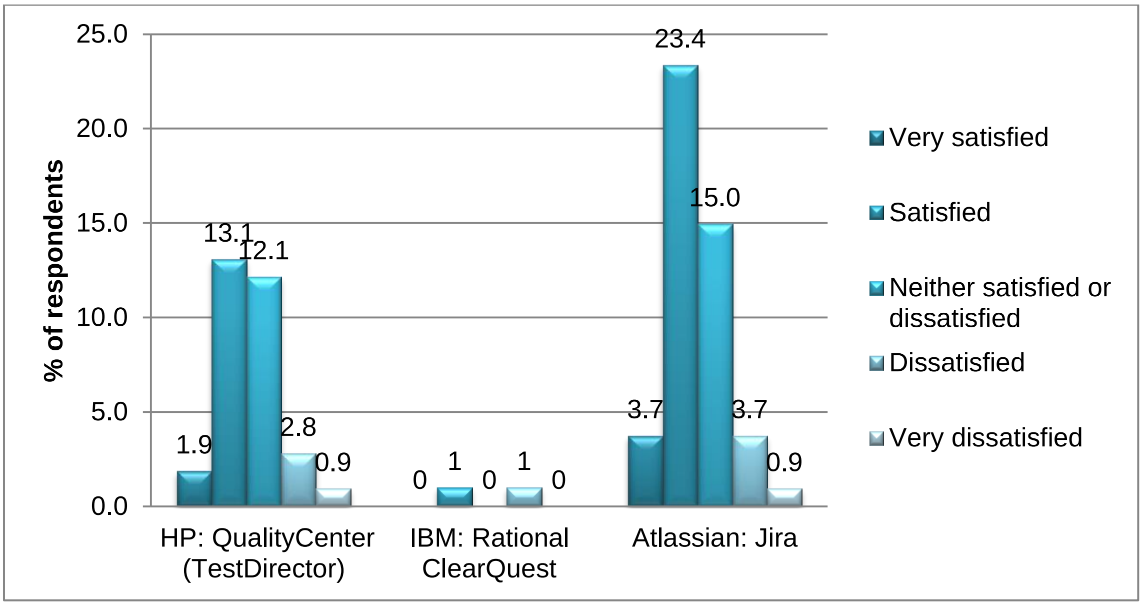


Figure 33; Evaluation of most popular tools

HP's Quality Center was evaluated as satisfied by 13,1 %. Atlassian's Jira got 23,4 % for satisfied and 15,0 % for neither satisfied or dissatisfied.

- Opinions about Incident management tools were divided; some of respondents were dissatisfied and some satisfied.
- Static analysis tool users were either satisfied or dissatisfied to MS Visual Studio tool.
- Selenium and IBM's tools were evaluated as equal; either satisfied or dissatisfied in test execution tools category.
- Absolutely most satisfied test management tools were Atlassian Jira.

- Jmeter was most used/popular tool in Performance/ load /stress testing tools category. See Appendix 2 for more on this topic. There was not found exactly this kind of evaluation from previous researches for comparation.

4.2.11 Analysing the comments from the respondents

In several questions was possible to give free comments also. In this section are analysed the ideas in the comments.

Most important message from respondents was that testing tools have to select to exactly to those needs what that the company or the project have. Every tool can be poor if it is not fulfilling the requirements for the project or company. In some companies already acquired tools are defining the future requirements; tools have to be compatible to each other has to be possible import data to the tool from the current testing tool. One reason that there are several tools used side by side is incompatible of the tools.

Testing over the web was seen slowly and this was hindering the testing, for example cloud services. Some people are thinking this is error although it is a feature. In the high-speed working environment the slowness is really disturbing the testing if you have to wait time after time the application to be ready for testing and that test engineer can continue the work.

In some open source testing tools was seen a challenge when vendors' developers have developed the tool as very technical that only company's developers can use it. Test engineers have realised the tool too technical and impossible to use, for example programming skills needed before possible to execute tests. But generally open source tools were seen as a good option for software testing tools.

In the tool selection process should think which functionalities are important for the executable work. Tool selection process is many times done on a higher level and selection criteria and understanding of executed work are different than test engineers.

Testing consultants are very often put out from testing selection process because they are seen as outsourced workers and not taken part in the important selection process itself. Opinions may have been asked but the selection process will be done inside the company. Therefore testing consultants have to use those testing tools which are available in the customer premises and this means various ranges of information and learning of different tools.

In smaller companies has to think carefully what testing tools and generally what resources will be invested. Testing tool investment and deployment costs are high. The price of the licenses and the tool itself are critical features for the decision maker. And many times it is a pity that “money talks” in the tool selection process. It is also possible that the business has not seen important to acquire testing tools, although testing teams would like to have those.

Testing tools can help the testing process significantly and this was seen also in the comments. Some had changed from a testing tool to another one and were happy after the change. Test management tool can make a huge difference in the testing process;

- The company’s testing process has become more clear
- Team members' roles and tasks have become more explicit
- Easier to follow the progress of the testing
- The connection between requirements and test cases makes easier to follow the test coverage.

Test management tools can be very complicated nowadays and they need continuous training after every upgrade.

Many testing professionals mentioned that they had to or have to study new tools and new functionalities by themselves. After the current testing tool has upgraded and new functionalities have established, training has not offered. That seems that new functionalities are not utilised enough and efficient way. Continuous training is needed for testing professionals.

Also was mentioned, that when consultants take an assignment it has seen as self-evident that the consultant has the skills to use all the testing tools. And if she/he has not, she/he has to learn very quickly the tools used in the company.

Many times self-learned testing tools habits can be very different from the company’s testing methods and processes. Many test engineers have then different usage habits with the tool. Therefore the learning process is time-consuming in the long run. It would be better to get the training in the very beginning of the project to learn the correct way to use the testing tool and according to the company's testing process and methods. Training leads the testing process to correct direction right away. Besides general testing tool training, the internal testing process training has to establish also.

Testing tools are used side by side in the different projects. The main reason for that are different needs, requirements and testing skills in the projects. Especially in the big companies there are differences in working habits between projects and teams inside the company. Test automation is not used in all projects and naturally therefore test automation tools are not always used. The company can be also international big corporate and the used software tools are different in international locations. Contrary to big projects and companies, small companies have different challenges; in the small project the establish the project based on the testing tool is more complicated and time-consuming than the project itself. There are projects where many suppliers are involved in the project, for example in development and testing. Therefore it is inevitable that several different tools are needed.

Sometimes companies can be stacked to own processes and working habits and are afraid of change and the consequences of it. The idea " that we have managed with the spreadsheet an word processing so far and we will also in the future". But this kind of company is not competitive. Old fashion thinking that 'the testing tool is not helping us' or 'we do not have time' can be also an obstacle for testing tool deployment in the projects.

The change from traditional life-cycle model to more agile or iterative life-cycle models are changing the role of testing and quality assurance in testing projects. Therefore new tools or functionalities are needed.

About current tools was mentioned that they do not support agile testing; they are not flexible to different life-cycle methods and processes. Also was commented that some tools have bugs in the production version which is hindering the working. Some tools require very high developing skills for example scripting skills, before you can use them; only developers can use them or have to have training for users. Some of the respondents were not convinced of features that one tool has all the possible functions; test management, test case creation, test execution, defect management, requirement management and automation. Automation tools were seen too technical for software testers, because needed developing skills. Needed functionalities for software tools were mentioned as better reporting, compatible with other testing softwares. These functionalities came up also in the survey responses. Sometimes testing tools have stayed behind the current development and are difficult to use for example in the performance testing. The user has to find a diversion to implement the wanted solution. In the performance testing seems to be a challenge that even vendor’s do not have an agreement of performance testing tool execution capability different testing environments.

As a summary, testing professionals planned to invest in the future to test automation tools, performance tools, modelling tools and test management tools. Current tools are not flexible to different file-cycle models and processes.

There are plenty of software testing tools available. Expertise is needed to select effective testing tools to to suitable areas of the testing process.

4.3 Results

4.3.1 Public report

The public report was created based on the poster of the Testing Days 2013. Some information was specified based on the feedback of the visitors. The public report includes for example the background information of the study, life-cycle answers, software tools information and finally the summary. The public report has been published in the Knowit web page in pdf-format (http://www.knowit.fi/Ajankohtaista/Pilvipalvelut-eivat-ole-viela-yleistyneet-testaustyokalujen-kaytossa/) or you can see it in Appendix 2.

4.3.2 Reports to customers

As results of this study, reports were created for customers of Knowit in English and in Finnish. The reports had a base from public report of the survey, but plenty of extra information was added.

The first section of the report is including the background information about the study and the survey including information about the respondent’s profession, age, education and industry. In the second section, current and future life-cycle models were covered. It was asked about which employees are using software testing tools and which kind of testing tools are in use and how the tools have been acquired.

A third section of the report concentrates on various testing tools. The software testing tools were classified as

- Test management tools
- Incident management tools
- Static analysis tools
- Test data preparation tools
- Test execution tools
- Test design tools
- The test harness / unit test framework tools
- Specialized unit test frameworks
- Dynamic analysis tools
- Performance/ load/stress testing tools

The report also gives answers to questions like; are the word processing and spreadsheet tools still in the use in the companies and purpose of the testing tools. Are they used for typical testing activities or something else? Companies' willingness to invest in new testing tools was asked next.

The fourth section of the report includes evaluations of different tools and vendors by respondents e.g. Test management tools and test execution tools. Finally, there is a summary of the whole report and conclusions. The customer reports are classified as confidential and not included in the study. The public report can be found in the Appendix 2.

4.3.3 Presentations

Preliminary results were presented by the author and instructor Kari Kakkonen in the morning seminar of the Knowit Oy, in the Kämp meeting, on the 24th of May 2013. The seminar was arranged for invited customers of Knowit Oy.

Secondly, a poster of the preliminary results was presented in the Testing Days 2013, in Otaniemi Espoo, on 4th of July 2013 by author. The poster was found a very interesting study by visitors.

Thirdly preliminary results were presented for personnel of Knowit, at the monthly meeting of Know It Oy, in May 2013.

5 Development plan

5.1 Proposal for life-cycle models

According to the survey the most popular life–cycle model at the moment is the scrum model. The second most popular was a V - model and third one iterative model. In the future the Lean and the Kanban life-cycle models seem to make biggest growth. Scrum and iterative life-cycle models will continue also very strong. V-model usage will be dropping continuously. Proposal for selecting a life-cycle model for a software project will be agile method Scrum, iterative, Kanban or Lean, because they seem to be most popular among respondents. It is recognised in the software business the shift from traditional life-cycle models to agile methods. The change to agile methods requires modifications in team structure and practises and need a plenty of automation.

5.2 Methods to acquire software testing tools

Many software tools are used side by side in the companies; commercial tools, open-source tool and proprietary tools. Normally testing tools are acquired by buying. Next used tools are open-source tools and self-coded. Only a minority of the companies were renting the tools. Maybe the most surprising result of this study was that the cloud services have not yet become a popular acquiring method in Finland. Reason for that can be many; incidents like cloud outages are beyond control, reliability of service is untrusted, continuous changes to the system are not easy to make, needs a lot of service level agreements, expert consultancy for the cloud may be missing, if needed new resources it is taking more time and delays in service. If there is a connection problem, these are also out of your hands; there is nothing you can do. The cloud affects the most fundamental attributes like performance, availability, security and scalability. Adapting new technologies, like cloud services takes time.

5.3 Recommendations for selecting software testing tools

There are plenty of software testing tools and they range from relatively simple to sophisticated software. New tools are being developed continuously to assist delivering the highest quality software needed for today’s high standard applications and complicated applications.

Software testing tools are essential to successful testing; those who are responsible for testing should be professional in executing software testing tools. The selected tools should be the most applicable to the environment in which the test engineers work and the particular types of software being tested. The individual who selects the test tools should also execute the test and known the tool well enough to use it efficiently. The selected tool should be reviewed and approved by the testing team, because the selected tool must be logical with the intension of the test plan. It should be selected the software testing tool not in separation, but connected to rest of the life-cycle tools and development observations. It should be noticed that most organizations and companies have a crossover of software tools. This multiplicity will not be just in the selection of software tools around of the life-cycles, but in testing itself; where different suppliers may have selected to support Web, mobile, package and legacy solutions. Different software tools should be selected to fit separate processes and testing skills. Although software tools are necessary, they do not solve fundamental faults in practises and lack of skills in the company. Therefore most of the companies have to improve their practises and methods before software testing tools start delivering top profit.

If the company fails with one tool, it does not mean that it will succeed with another. Some corporations have tested many tools and failed in the same way with each one. Unfortunately, the tools and or the personnel are regularly accused for the failure, as a matter of fact the software testing tool acquiring project was not effectively designed and managed.

By supporting the personnel is the best way to utilize the software testing tools. The test engineers and persons who will be using the tools should have a possibility to affect the usage of the tool. The infrastructure should build and placed around to support testing team.

Whatever software tool the company have, the training is very important. Personnel who will be using the tools should have training in the early phases of the test tool implementation process. It is possible to acquire supplier courses or online tutorials. Effective training can avoid a plenty of wasted time.

The market in the software testing is taking steps forward to better support packaged applications. However, new technologies are missed also because they are at the base of cloud and merged applications. The change from traditional life-cycle methods to Agile methods is another interesting driver in the software testing market. Agile method has been spread from software developers to include the whole software testing team. This affects software testing teams by adding effort for progress of test automation. Not only progressing of software testing but also the managing of software test environments and the release of software into the software test environments. This kind of change will generate a selection of testing styles that are balanced to business types.

Software testing may be a very expensive process, but poor software quality leads to plenty of difficulties as dissatisfaction among users, increased development work and maintenance costs. Therefore, it is important to have beneficial and carefully selected set of software testing tools and methods in order to have best possible quality of software. And in the end, the quality efforts will have a positive effect on the bottom line of the business.

There are traditionally three methods of competing in the testing tools market; price, service and features. I support Gil Bloom’s idea to compete with solution and concentrate on customers' challenges and problems rather than feature lists. If the business is concentrated around feature list, it will soon become overwhelming and impossible to handle and impossible to make everyone happy. In a business based on feature lists, some of the end users will inevitably be happy when seeing their wish list but however the majority will be disappointed. When the business is a solution-focused the feature list is less important and the company can direct limited resources towards developing the simplest fix. Then the company can control the costs and more easily compete with the price. Being solution-oriented is not an easy task, but for customers and therefore the suppliers, are worth it. (Bloom, G. 2013)

5.4 Recommendations for software testing tool development

The price still seems to be the most significant reason for selecting software testing tools. Small companies cannot afford to acquire expensive licenses and tools. It might be a good idea to produce light version of popular tools for small and mid-sized companies.

The second thing the vendors have to pay attention to is the user friendliness of the tool. This was the second important functionality according to the survey. It is understandable if the tool takes a lot of expensive training and it is not logical to use, it will be less used or maybe totally turned down by the users. Gartner has researched 79 % of the allocated software project budget is used for testers and tool training. Secondly most important features of the tool were seen functionality and management functions.

It seems that many suppliers are facing challenges in keeping up with the speed of technological change, because of new browser releases, mobile devices or UI technologies. New customer capabilities and the fusions need to test on several platforms. Suppliers should put more effort for progress of software testing tools if they want to be top in the market. Most of the testing tools are still supporting more traditional life-cycle models and development for Agile life-cycle model testing tool should be increased.

It also seems that direct marketing expertise is becoming less important. Instead of that community feedback and price are more significant qualities. It will be a challenge for new organisations to maintain high levels of innovation and at the same time build robust execution.

6 Conclusions

This section summarises the overall research process, presents the key recommendations based on the results, considers the reliability and validity of the results and provides recommendations for further studies.

6.1 Summary and discussion

The purpose of the present study was to analyse the use of testing tools, to identify which functionalities affect to testing methods selection criteria. The planned outcome of the study was a information of software testing tools, recommendations which kind of testing tools to use and used selection criteria and reports to Knowit Oy for customer needs.

For a background study the best available material from literature was researched to find information about used testing tools and their purpose in software testing. Also the basic idea of software testing was presented; Questions like ‘why the testing is important’ and ‘how the quality and found defects influence to the quality’ were handled. Possible influences can be costs derived from low quality software and maybe companies ending up losing their reputation. Life-cycle models used in software testing were shortly introduced.

The object of the study was based on ISTQB classification. A selection of software testing tools was made. It was not possible to take under investigation all testing tools and therefore selection was made. Used testing tools classification was taken from ISTQB. The selected testing tools were shortly presented along usage descriptions and the use of them described.

Next the current software tool selection process was presented including which methods was used. The very same process was also repeated to test automation tools. Benefits and risks using testing tools have to take into account when selecting testing tools. Risks and benefits of test automation were also covered.

After literature research was done, it was time to conduct the survey. Questions were planned together with my instructor Kari Kakkonen and help of colleagues in Knowit Oy. Several review rounds were established. The questions were planned to a web - form and send by the Digium enterprise application and to TestausOSY members by mailing list. The survey was sent in the February of 2013. After two weeks time period for answering the survey, the data was taken out to closer analysis into MS Excel spreadsheet, because I did not have access to the Digium enterprise tool. After studying the data, I went through the data and produced graphs, tables and PowerPoint presentations based on findings from the information. Preliminary results were presented in Know it’s monthly meeting in May, Know it’s a morning seminar in June and in the Testing Days 2013 in Otaniemi Espoo in June 2013. Finally the public report was published in Knowit internet page on September 2013 and the winner of the draw was also published. The end result of the study was two reports for Knowit’s customer purposes in English and in Finnish, which Knowit will deliver forward to selected customers.

Technology is evolving much faster than software testing organizations, making it difficult to test many emerging technologies and devices. A shift from traditional life-cycle models to Agile methods is recognised in the software business. The change to Agile methods requires modifications in team structure and practises and testing is heavily based on need of plenty of automation.

Testing tool selection process is very complicated and time consuming. It has to be organised systematically. A separate project team for test tool selection would be ideal. This team should include also testing people and especially test engineers who are really going to use the tool.

Testing tools are acquired most commonly by buying. Next comes open-source tools and proprietary tools acquiring methods. Cloud services have not yet gained popularity according to this study. However all these tools are used side by side in the companies including word processing tools and spreadsheet tools.

6.2 Practical next steps

This study concentrates to investigate the use of software testing tools in Finnish IT-companies. First, background information was investigated from respondents like age, profession, education, location, size of the company/ customer company and an operating area of the company. In addition, used life-cycle models were investigated; the current life-cycle models in projects and future trends. Under investigation was also, which employees are using software testing tools and the survey result was that testing professionals. Testing professionals gotten training was also investigated. The survey also gave answers to that which way the software testing tools are acquired and what kind of tools there is in use in the companies. The purpose of software testing tools was also presented. Finally, there is presented the results of a popularity of 127 different software tools and their evaluation. All together 61 questions were asked; 14 general and 51 testing related questions.

After this study the Knowit Oy has the survey questions and results in a Digium enterprise system for later use if needed. The logical next step would be to repeat the research for comparison and benchmark after two years. Database of the results is also in an MS Excel spreadsheet and they are archived for later use or analysis to Knowit database.

Public report was published and is currently available in the web page of Knowit Oy. Two confidential reports in English and in Finnish are available to be delivered to the customers of Knowit Oy, if needed.

Logical option for future research would be to investigate why the cloud services have not reached popularity as testing tool acquiring method. This may have been the most interesting result of the research that cloud services were used only 6 % as acquiring method.

There is also another interesting research area like test data management in customers; problems and success stories. Test data management has become very challenging in complicated and big data environments and is always problematic for test engineers. Other interesting topics would be to research how open source and commercial tools work side by side, by qualitative research methods such as tool user interviews with selected companies or to research how test tools are used, by qualitative research.

This study is already utilized in practise by following ways:

- Selected survey results will be used in Knowit media channels (newsletter, website, Facebook)
- Knowit customers will receive the customer report followed by a consultant-lead discussion of further test automation possibilities
- Knowit partner selection will be reconsidered based on the most popular software testing tools identified in the study.
- Test automation productizations will be further developed utilizing information gotten from the survey.
- Recommendations of life-cycle models and tool selection will be utilized in consultant-lead discussions with customers.

6.3 Evaluation of the study

6.3.1 Outcome/ Objective

The outcome of the study was three different reports; two confidential reports; in English and in Finnish and one public report were published. The public report was very wanted especially in the Testing Days 2013 in Otaniemi, Espoo and many visitors wanted the report before published officially on the webpage.

Confidential reports will be delivered to customers of Knowit in consultant-lead discussions. Reports have fulfilled preliminary expectations.

One object of the study was to give information of used testing tools in Finland. The survey gave valuable information for example of commercial, open source and proprietary tools. Most popular software testing tools are presented and recommendation of them will be used in customer discussions.

It is clearly seen, that software projects are moving into a more iterative direction in life-cycle models. This study also supports this idea. In the future the for example Kanban and Lean life-cycle models are expected to be more popular methods.

In this study is presented also recommendations for tool development. The study clearly brings forth that the price has to be thought very carefully to compete in the market. Small companies cannot buy not at all expensive licences. Most important functionalities to the customer/user according to this study among price were tool functionality and management of the tool. Only the future will show is the tool development business developing their business to this recommended direction.

The study helps concentrating training efforts of most used testing tools, ensuring the up-to-date knowledge of testing tools for customers and for consultants of Knowit. Results are also utilized by marketing department in customer-lead discussions.

Based on the study, preliminary results of the survey and the presentations were established;

- Morning seminar of Knowit in May 2013
- Knowit monthly meeting in May 2013
- Testing Days 2103, In Espoo, Otaniemi in June 2013

6.3.2 Qualitative and quantitative research

This section covers the definitions of qualitative and quantitative researches, the meaning of them and which category this survey belong and why.

The qualitative research consists of several traditions, approaches, material collection methods and analyse methods. Therefore, it is not any specific science research method or just one way to do research. The name ‘qualitative research’ can be criticized, because it can be misleading and can give the impression of better, the more soft research compared to superficial and hard quantitative research. All researchers are in matter of fact just scratching the surface. With the help of the research, it can never reach entirely a phenomenon and total deep of it. But by proper planning and implementation, by repeating the research and approaching the phenomenon from a different perspective it is possible to get diverse information and therefore to increase understanding related to the phenomenon. (Saaranen-Kauppinen, A. et al. 2006).

One of the milestones of the qualitative research is the theory of the observations. By the theory of the observations are meant the methods which are affecting to end results. To research end results may affect following;

- The researcher's opinion of the phenomenon
- What kind of meanings is given to the phenomenon
- What kinds of methods are used in the research. (Tuomi, J. et al. 2009)

Research results cannot be disconnected from the used observation methods or users. There is no objective information; all the information is seen as subjective because the researcher has decided the research frame according to his/her own understanding. Considering this idea, arguments of qualitative research highlight the theory as a base for all research. As a summary, the qualitative research cannot adopt the theory totally to itself, but it cannot also abandon it. (Tuomi, J. et al. 2009)

According to Tuomi and Sarajärvi, It is more important the researcher to know what he/she is really doing than define the used methods. It is not all about technical methods of research, but also used ethical methods; researcher has to have a clear objective of the research. No matter of the all given instructions, every research has its own ethical problems. (Tuomi, J. et al. 2009)

Surveys are generally considered as quantitative research, because that is adapting statistical methods. Surveys are mainly consisting of measured figures and numbers, although questions are presented by written, but the results are presented with numbers and figures. In addition, by written is given additional information or answers which would be impractical to present by numbers

(Vehkalahti, K. 2008). This study has categorised as quantitative research because its' nature of numbers and figures. Also the survey has done to dozens or hundreds of people, not by individual interviews.

6.3.3 Validity and reliability

This section consists of the self-evaluation and analyse the validity and reliability of the study.

Validations is answering to question are we measuring correct things? This is the first criteria for quality. (Vehkalahti, K. 2007) And on the other hand; is it investigating what has promised in the beginning of the research? (Tuomi, J. et al. 2009)

There is always some uncertainty in researches because of the ways the data are collected and measured. The results are dependent of the sample group and measurement methods. This study is based on the survey which was made for software testing professionals by web-form. The target group for the survey was IT and testing professionals in Finland. It is not possible to investigate the whole population of testing professionals. The Natural choice was to select Knowit's contacts as a sample group and for second TestausOSY society mailing list. Third contact list was Solidiet registration. The sample group contacts had to go through and remove those ones who are not related to software testing. The goal was to get a good selection of responses from various industries in Finland; small business companies and large companies and everything between that. The research accomplishes this objective quite well. But because 84 % of respondents were from south of the Finland, the study cannot seen to represent the whole population of Finland.

The second contact source was a TestausOSY society mailing list. This email list is open to anybody. It is possible that there are people who are not possibly IT or testing professionals. The second unknown issue in this mailing list is that how much there was overlapping with Knowit's contact list and Solidiet registration list. It was not possible to solve beforehand because for example in the TestausOSY mailing list is only known the given email address, not any other information about the creator of the account.

The reliability is defined as the ratio of the true variance of the total variance of the measurement. The true variance does not include the variance of the random measurement error (Vehkalahti, Puntanen, Tarkkonen 2006). Among that Tuomi and Sarajärvi emphasizing the repeatable of the research. There are no explicit instructions how to estimate the reliability of the research. The research will be estimated as a whole and the consistency of the research. (Tuomi, J. et al. 2009)

The aim of this study was to measure software testing tool usage in Finland and to identify methods selecting testing tools. The purpose was to implement recommendations how to select software testing tools, which life-cycle methods to use, recommendation of tool selection and software tool development. Also reports were established for customer purposes. Many remarkable results were received and evaluation of the tools. Recommendations were done as seen in the chapter 5. Overall there was handled 127 different software testing tools in the study. The results were very interesting and generated plenty of valuable information. Graphs and final reports were created mainly for the need of Knowit Oy. Reports have fulfilled preliminary expectations of Knowit and author.

The subject of the study is important and interesting for me as a testing professional. Also because of my employer's interest towards the topic was a crucial reason to make this study. There were no remarkable presumptions in the beginning of the study of the results; I have seen in many projects the change from traditional software life-cycle models for more agile models. In the usage of the software testing tools was the idea that some leading vendors may be popular, but there were many surprises related to open source tools. The results in the end were speaking for themselves.

The process of working with the study has taught me different testing tools and selection processes. This has given me understanding of the problems and difficulties in testing tools selection process. Also this has taught me that the selection process has to do with time and systematic way. The process of the study has also taught me the method to plan survey researches. There are plenty of software testing tools. Expertise is needed to select effective testing tools for suitable areas of testing processes.

References

Abraham, C. (n.d.). *Test Automation Tool Evaluation*. Aspire Systems.

Ahn,Y., Sampath, V. (2012). ISO/IEC JTC1/SC7/WG4 Standard Software engineering – Capabilities of software Testing Tools.

AP test. Software testing specialist (n.d.). *Software QA Testing and Test Tool Resources.* Available at http://www.aptest.com/resources.html#reqs [Accessed 23 Sep 2012].

Bach, J. (2012). *James Bach's blog.* Available at http://www.satisfice.com/blog/archives/118 [Accessed 3 of March 2013].

Bhuvaneswari, T., S Prabaharan, S. (2013). *A Survey on Software Development Life-cycle Models.* IJCSMC, Vol. 2, Issue. 5, May 2013, p. 262 – 267.

Black, R. (2002). *Managing the Testing Process.* 2nd edition. Wiley Publishing, Inc.

Black, T. (2011). *20 Essential Tools for Project Management.* Available at http://www.inc.com/guides/201102/20-essential-tools-for-project-management.html [Accessed 11 March 2013].

Bloom, G. (2013). *Competing For Solution; New Approach for Testing Tools.* Sponsored by Testuff Ltd. EuroSTAR Software Testing Community.

Buksh, J. (2011). Performance testing definitions. Available at http://www.perftesting.co.uk/performance-testing-definitions/2011/08/17/ [Accessed 14 Jan 2013].

Chatzoglou, P., Macaulay, L. (1996). *Requirements Capture and Analysis A Survey of Current practise.* Requirement Engineering Journal 1 (2).

Chernenko, I. (2011). QATesLab;*7 ways to measure test coverage.* Available at http://blog.qatestlab.com/2011/03/31/7-ways-to-measure-test-coverage/ [Accessed 3 March 2013]

Crispin, L., Gregory J. (2009). *Agile Testing; Practical guide for Testers and agile teams.* Boston: Pearson Education, Inc.

Easterbrook, S. (2012). *Lecture 19: Static Analysis Tools.* University of Toronto Department of Computer Science.

Fewster, M., Graham, D. (1999). *Software Test Automation. Effective use of test execution tools.* Pearson Education Ltd. Great Britain.

Garfinkel, S. (2005). *History's Worst Software Bugs.* Available at *http://www.wired.com/software/coolapps/news/2005/11/69355?currentPage=all* [Accessed 27 Jan 2013].

Glenford J. Myers (2004). *The Art of Software Testing, Second Edition*. John Wiley & Sons, Inc., Hoboken, New Jersey.

Graham, D., Fewster, M. (2012). *Experiences of Test Automation. Case Studies of Software Test Automation.* Addison-Wesley. Boston.

Heppenstall, D. (2009). *The Ultimate Goal of Software Engineering.* Available at http://www.heppenstall.ca/index.html [Accessed 27 Jan 2013].

Hoffmann, M., Kühn, N., Weber, M. (2004). *Requirements for Requirements Management Tools*. Daimler Chrysler AG Research & Technology. Berlin.

Iconma. Testing methodology. Available at http://www.iconma.com/ser_testing_methodology.htm [Accessed 1 November 2013].

Ilchenco, A. (2011). *Useful Tips for Choosing a Test Automation Tool .* Available at http://blog.qatestlab.com/2011/06/08/useful-tips-for-choosing-a-test-automation-tool/ [Accessed on 11 March 2013].

Illes, T., Herrmann, A., Paech, B., Rückert, J. (2005). *Criteria for Software Testing Tool Evaluation – A Task Oriented View.* Institute for Computer Science, University of Heidelberg, Germany.

ISO/IEC JTC 1/SC 7 Standard. *Software and systems engineering.* Secretariat: SCC, Canada.

ISTQB (2011). *Advanced Level Syllabus - Technical Test Analyst.* Available at <http://www.istqb.org/downloads/viewcategory/16.html> [Accessed 22 Sep 2012].

ISTQB (2011). *Advanced Level Syllabys - Test Manager.* Available at <http://www.istqb.org/downloads/viewcategory/16.html> [Accessed 22 Sep 2012].

ISTQB (2011). *Advanced Level Syllabus - Test Analyst.* Available at <http://www.istqb.org/downloads/viewcategory/16.html> [Accessed 22 Sep 2012].

ISTQB (2011). *Expert Level Syllabus - Improving testing process.* Available at <http://www.istqb.org/downloads/viewcategory/16.html> [Accessed 22 Sep 2012].

ISTQB (2011). *Expert level Syllabus - Test management.* Available at <http://www.istqb.org/downloads/viewcategory/16.html> [Accessed 22 Sep 2012].

ISTQB (2011). *Foundation Level Syllabus (2011).* Available at <http://www.istqb.org/downloads/viewcategory/16.html> [Accessed 22 Sep 2012].

ISTQB (2011). *Glossary of testing terms.* Available at <http://www.istqb.org/downloads/viewcategory/16.html> [Accessed 22 Sep 2012].

ISTQB (2012). ISTQB Certification – Foundation Level syllabus Exam Certification.com. Available at http://istqbexamcertification.com [Accessed 16 Nov 2012].

Johnson, D. W., Dev Bistro (n.d.). *Evaluating Testing Software & Tools.* Available at http://www.devbistro.com/articles/Testing/Evaluating-Testing-Software-Tools [Accessed on 29th Sep 2013]

Johnson, J., et al. (1994). Chaos: Charting the seas of Information Technology. Published report. The Standish group.

Kaner, C., Bach, J., Pettichord, B. (2001). *Lessons Learned in Software Testing. A Context-Driven Approach.* John Wiley & Sons Inc. New York.

Khaled, M. M., Al-Qutaish E. R., Muhairat I. M. (2009). *Classification of software testing tools based on the software testing methods.* Al-Zaytoonah University of Jordan.

Kaushalam, Information Technology architect. *QA/Testing Review.* Available at http://www.kaushalam.com/qa-testing-review.html [Accessed 30 October 2013].

Khannur, A. (2011). *Software Testing - Techniques and Applications.* Pearson Publications, Bangalore, India.

Koomen, T., Pol, M. (1999). *Test Process Improvement. A practical step-by-step guide to structured testing.* Addison-Wesley. London.

Knowit Oy, (2012). *Business, mission, vision and goals.* Available at <http://www.knowitgroup.com/About-us/Business-mission-vision-and-goals/> [Accessed 25 May 2012].

KvantiMOTV - Menetelmäopetuksen tietovaranto. Tampere: Yhteiskuntatieteellinen tietoarkisto. Available at http://www.fsd.uta.fi/menetelmaopetus/ [Accessed 25 May 2012].

Lewis, W.E. (2004). *Software Testing and Continuous Quality Improvement.* 2nd edition. Auerbach publications: New York.

Lake, M. (2010). *Epic failures: 11 infamous software bugs.* Available at *http://www.computerworld.com/s/article/9183580/Epic_failures_11_infamous_software_bugs?taxonomyId=18&pageNumber=6* [Accessed 27 Jan 2013].

Larman, C. (2007). *Agile & Iterative Development; A Manager's Guide.* Massachusetts: Pearson Education, Inc.

McConnel, S. (1996). *Rapid Development.* Redmond: Microsoft Press.

McDonald, M., Robert Musson, R., Ross Smith, R. (2007). *The Practical Guide to Defect Prevention.* Microsoft Press.

Murphy, T. E., Wilson, N. (2013). *Magic Quadrant for Integrated Software Quality Suites.* Gartner.

Nguyen, Q.H., Johnson, B., Hackett, M. (2003). *Testing Applications on the Web.* 2nd edition. Indianapolis: Wiley Publishing, Inc.

Pingdom (2009). *10 historical software bugs with extreme consequences.* Tech blog. [Accessed 6 March 2013] http://royal.pingdom.com/2009/03/19/10-historical-software-bugs-with-extreme-consequences/

Pinkster, I., van de Burgt, B., Janssen, D., van Veenendaal, E. (2006). *Successful Test Management.* Springer: Berlin.

Poston, M.R., Sexton, P.M. (1992). *Evaluating and Selecting Testing Tool.*

Price-Jones, N., (n.d). *Automated Test Tool Selection.* NVP Software Testing. Simplilearn.com. Available at http://www.simplilearn.com/ [Accessed 11 Jan 2013].

Rajander-Juusti, R. (2012). *Miten toteutan toimivan verkkokyselyn?* Helsinki: Questpack.

Rothman, J. (2000). *What Does It Cost You To Fix A Defect? And Why Should You Care?* Available at http://www.jrothman.com/2000/10/what-does-it-cost-you-to-fix-a-defect-and-why-should-you-care/ [Accessed 11 Oct 2013].

Rouse, M. (2011). *Performance testing.* Available at http://searchsoftwarequality.techtarget.com/definition/performance-testing. [Accessed 11 Jan 2013].

Saaranen-Kauppinen, A., Puusniekka, A., (2006). KvaliMOTV - Menetelmäopetuksen tietovaranto. Tampere: Yhteiskuntatieteellinen tietoarkisto. Available at http://www.fsd.uta.fi/menetelmaopetus/ [Accessed 30 Nov 2013].

Sasankar, A. B., Chavan,V. (2011). *Survey of Software Life-cycle Models by Various Documented Standards.* IJCST Vol. 2, Issue 4, Oct . - Dec. 2011.

Sommerville,I. (2011). *Software Engineering*. Addison-Wesley. New York.

Srivatsa, H., M Kumar, G.M., Presley, S. (n.d.). *Agile Project Management of non-collocated teams.* Available at http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.90.1950&rep=rep1&type=pdf [Accessed 16 Nov 2012].

Surveymonkey (n.d.). *Kyselytutkimuksen otoskoko.* Available at https://fi.surveymonkey.com/mp/sample-size/ [Accessed 24 October 2013].

TechTarget; Search win development (n.d.). *Static analysis.* http://searchwindevelopment.techtarget.com [Accessed 16 Nov 2012].

Testhouse (n.d.). *Test Process Optimisation Test Tool Selection.* Available at http://www.testhouse.net/test-tool-selection/ [Accesses on 12 Sep 2013].

Tuomi, J., Sarajärvi, A. (2009). *Laadullinen tutkimus ja sisällönanalyysi*. Tammi: Helsinki.

TJ Tools Journal (n.d.). *Top Open Source Test Management Tools.* Available at http://www.toolsjournal.com/testing-lists/item/263-top-open-source-test-management-tools [Accessed 22 Sep 2012].

TJ Tools Journal (n.d.). *Top 15 Performance Testing Tools.* Available at http://www.toolsjournal.com/tools-world/item/156-top-performance-testing-tools [Accessed 22 Sep 2012].

TestausOSY. Finnish Association of Software testing. Available at http://testausosy.fi/ [Accessed 01 Nov 2013].

Usertesting.com (n.d.). *10 Tools Every User Experience Pro Should Know About.* Available at http://www.usertesting.com/blog/2013/11/07/10-tools-every-user-experience-pro-should-know-about/ [Accessed 24 October 2013].

Van Veenendaal, E., Graham, D. (2008). *Foundations of Software Testing: ISTQB Certification*. Cengage Learning EMEA, London.

Vector Software, Inc. (n.d.) *How to Evaluate Embedded Software Test Tools Vector Software.* Available at http://www.vectorcast.com/pdf/white-papers/how-to-evaluate-embedded-software-test-tools_vector_wp.pdf [Accessed 24 March 2013].

Vehkalahti, K. (2007). *Kyselytutkimuksen menetelmät ja mittarit.* Tammi. Helsinki.

Vehkalahti, K., Puntanen, S., Tarkkonen, L. (2006). *Estimation of reliability: a better alternative for Cronbach's alpha.* Department of Mathematics and Statistics, University of Helsinki, Finland.

Wieczorek, M., Meyerhoff, D. (2001). *Software Quality; State of the art in management, testing and tools.* Springer. Köln.

Woods, D. (2005). *What Is Open Source?* Available at
http://www.onlamp.com/pub/a/onlamp/2005/09/15/what-is-opensource.html.
[Accessed 24 October 2013].

Yang, Q., Jenny Lee J., Weiss, D. M. (2007). *A Survey of Coverage-Based Testing Tools.* Published by Oxford University Press on behalf of The British Computer Society. Avaya Labs Research, USA.

Software testing tool survey questions

Software testing tools survey

Welcome to take part in the software testing tools survey

You are welcome to take part in the software testing tools survey, in which we will investigate what are the most common software testing tools used by testing professionals in Finland.

The survey will be done anonymously and confidentiality is guaranteed. It will take only 5 to 10 minutes to answer the survey. It is important that you will answer to all questions; otherwise the system will not let you to continue.

Answer the questions by clicking the mouse to the correct option for you and/or follow the instructions given in the question. By selecting the ‘Continue’ –button you will get forward in the survey and to the next page. Back to the previous page you will get by selecting the ‘Back’ –button.

Address and name information is optional and used only to take part in the draw. Address, name and email information is not used for marketing purposes. We will use SSL-secured connection, which will guarantee the safety of survey information. It is confidential to answer to the survey and it is not possible to recognize individual answers. Information will not be forwarded to a third party. If you have any questions don’t hesitate to ask by email minna.tiitinen@knowit.fi

Best regards,

Knowit Oy

Source for addresses:

Knowit Oy's contact register
Tehtaankatu 27-29D, 00150 Helsinki, Business ID 1053026-7

TestausOSY

Soliditet register of Bisnode Finland

1. Common questions (1/2)

* A response to this question is required

1.1 Gender of respondent

- Male
- Female

1.2 Age of respondent

- Below 20
- 21- 30
- 31-40
- 41-50
- 51-60
- Over 61

1.3 Education (select the highest you have)

- Interimediate school
- High school/ Secondary school graduate
- Professional education

- University of applied sciences
- Higher education institution
- Licentiate or doctorate
- Other, what?

1.4 Your profession/role in the company

- Test automation specialist
- System tester
- Developer
- Business unit representative
- Requirement engineer / business analyst
- Project manager
- Scrum master
- Software architect
- Software manager
- Performance tester
- Test lead/ Test manager/ Test coordinator
- Outsourced testing professional
- Unemployed
- Other, what?

2. Common questions (2/2)

* A response to this question is required

2.1 Size of the company (employees)

- 1-10
- 11-50
- 51 – 100
- 101 – 500
- 501 - 1000
- Over 1001 employees
- I don't know

2.2 Company’s operating area

- Science or technical area
- Administration and support
- Information and telecommunication
- Governance and national defence
- Mining
- International organization and commission
- Real estate business
- Teaching
- Warehouse and transfer
- Agricultural, forestry, fishing business
- Hotel and restaurant business
- Other service
- Finance and insurance business
- Constructing business
- Electricity, gas, heating service and cooling system business
- Art, show and entertainment business
- Industry
- Health care and social service

- Business area unknown
- Retail business
- Environmental business
- Other, what?

2.3 Location of the company

- Åland province
- South of Finland province
- East of Finland province
- West of Finland province
- Lapland province
- Oulu province

2.4 The company has…

- Own testing department
- Outsourced testing department
- Both, own testers and consultants
- Something else, what?

2.5 Which way the testing tools are acquired?

- Bought
- Cloud service
- Rented
- Open Source
- Coded by company itself
- Another way, what?

3. Questions dealing with software testing

* A response to this question is required

3.1 What is a life-cycle model of your current projects?

- Cleanroom
- Crystal methods
- DSDM
- Dual Vee model
- Extreme programming (XP)
- Feature driven development
- Incremental
- Iterative
- Kanban
- Lean
- Prototype model
- RAD
- RUP
- Scrum
- Spiral
- TDD
- V-model
- Other, what?

3.2 What is the life-cycle model you think you will use in the future in your company?

- Same life-cycle model as before
- I don’t know
- Changing to some other model, which?

3.3 Which life-cycle model will be a future trend in your opinion?

- Cleanroom
- Crystal methods

- DSDM
- Dual Vee model
- Extreme programming (XP)
- Feature driven development
- Incremental
- Iterative
- Kanban
- Lean
- Prototype model
- RAD
- RUP
- Scrum
- Spiral
- TDD
- V-model
- I don't know
- Other, what?

3.4 Which employees of your company are using software testing tools?

- Test Automation Specialists
- System testers
- Developers
- Business people
- Requirement engineers / business analysts
- Project manager
- Scrum master
- Software architect
- Software manager
- Performance testers
- Test leads/ Test managers/ Test coordinators
- Outsourced testing professionals
- Others, who?

3.5 Is your company using commercial or open source software testing tools?

- Commercial software testing tools
- Open source testing tools
- Proprietary tools
- Something else, what?

4. Questions dealing with software testing

* A response to this question is required

4. 1 What is a life-cycle model of your current projects?

- Cleanroom
- Crystal methods
- DSDM
- Dual Vee model
- Extreme programming (XP)
- Feature driven development
- Incremental
- Iterative
- Kanban
- Lean
- Prototype model
- RAD
- RUP

- Scrum
- Spiral
- TDD
- V-model
- Other, what?

4.2 What is the life-cycle model you think you will use in the future in your company?

- Same life-cycle model as before
- I don’t know
- Changing to some other model, which?

4.3 Which life-cycle model will be a future trend in your opinion?

- Cleanroom
- Crystal methods
- DSDM
- Dual Vee model
- Extreme programming (XP)
- Feature driven development
- Incremental
- Iterative
- Kanban
- Lean
- Prototype model
- RAD
- RUP
- Scrum
- Spiral
- TDD
- V-model
- I don't know
- Other, what?

4.4 Which employees of your company are using software testing tools?

- Test Automation Specialists
- System testers
- Developers
- Business people
- Requirement engineers / business analysts
- Project manager
- Scrum master
- Software architect
- Software manager
- Performance testers
- Test leads/ Test managers/ Test coordinators
- Outsourced testing professionals
- Others, who?

4.5 Is your company using commercial or open source software testing tools?

- Commercial software testing tools
- Open source testing tools
- Proprietary tools
- Something else, what?

5. Used Test Management Tools

5.1 What Test management tools do you have in the company?

- Borland: SilkCentral TestManager
- HP: QualityCenter (TestDirector)

- IBM: Rational ClearQuest
- Jira
- Jite (open source)
- MKS: MKS Integrity for Test Management
- Mozilla Testopia (open source)
- Oracle (entinen Empirix)
- qaManager (open source)
- QaTraq Professional (open source)
- Smartbear: QAComplete, ALMComplete
- Tesly (open source)
- Test Rail
- TestLink (open source)
- TETware (open source)
- Trello (open source)
- WebTst (open source)
- XQual Studio (open source)
- Other, what?

5.2 What Incident management tools do you have in the company?

- Atlassian: JIRA
- Bugzilla (Open source)
- IBM Rational ClearQuest
- Mantis
- PVCS Tracker
- Roxagi: Prtracker
- Other, what?

5.3 What Static analysis tools do you have in the company?

- TEST
- BoundsChecker
- C++test
- CodeHealer
- CodeSurfer
- FindBugs
- Flawfinder
- HP Fortify Static Code Analyzer (SCA)
- IBM: Rational Software Analyzer
- JTest
- LDRA Testbed
- Microsoft Visual Studio
- PC-Lint/FlexeLint
- Polyspace
- PVS-Studio
- QA C/C++
- Rational Logiscope
- Rational Software Analyzer
- SofCheck Inspector
- Sonar
- Syhunt Code
- Other, what?

5.4 What Test data preparations tools do you have in the company?

- Compuware: File-AID
- HP Test Data Management software
- GridTools
- IBM Princeton Softec Optim
- Oracle Data mining

- Rational Quality Manager
- Visual Studio
- Other, what?

5.5 What Test execution tools do you have in the company?

- HP: QuickTestProfessional
- Borland: SilkTest
- IBM: Rational Functional Tester, Rational Robot, Rational Tester
- IBM: Telelogic Tester
- Compuware: TestPartner
- Oracle (previous Empirix)
- AutomatedQA: TestComplete
- Bredex: GUIDancer
- Parasoft: Jtest, C++Test, SOATest
- OpenSTA (Cyrano) (TestCom¬mander + ScriptModeler + TestExecuter + Repository)
- Eviware: SoapUI Pro
- Watir (IE:lle, opensource)
- FireWatir (Firefoxille, opensource)
- FitNesse (open source)
- Selenium
- Smartbear: TestComplete
- Robot Framework
- Cucumber
- Concordion
- Other, what?

5.6 What Test design tools do you have in the company?

- Conformiq Designer
- Leirios
- Hexawise
- AllPairs.java (freeware)
- HP: QualityCenter (TestDirector)
- Telelogic: Modeler, Statemate
- IBM: Rational Manual Tester
- Smartbear: QAComplete
- Conformiq Designer/Modeler
- ALLPAIRS
- CTE
- Other, what?

5.7 What Test harness/ unit test framework tools do you have in the company?

- xUnit family
- JUnit
- CppUnit
- NUnit
- DUnit
- Unit++
- GNAtest
- TestNG
- JTest
- Sonar
- Unitils
- SpryTest
- Visual Studio Unit Testing Framework,
- MbUnit
- .TEST
- csUnit

- C/C++test
- Other, what?

5.8 What Specialized unit test frameworks tools do you have in the company?

- DejaGnu
- WebInject
- JBoss Test Harness
- JavaTest harness (jtreg)
- Cactus, Jakarta
- Mockito
- JBehave
- Concutest
- JMock
- JDave
- Other, what?

5.9 What Dynamic analysis tools do you have in the company?

- Advanced .NET Tool Suite (ANTS)
- AQtimePro
- BoundsChecker
- Compuware: DevPartner Studio
- DevPartner Studio
- DevPartner Studio
- IBM: Rational Purify
- Insure++
- Parasoft: .Test
- Parasoft: Insure++
- Rational Purify
- Security Innovation: Holodeck
- Smartbear: AQtimePro
- Other, what?

5.10 What Performance/ load/ stress testing tools do you have in the company?

- AutomatedQA: AQTime
- Benchmark Factory (DB)
- Borland (Microfocus): SilkPerformer
- Grinder
- HP: Performance Center (LoadRunner /Performance Center), Diagnostics)
- IBM: Rational Performance Tester
- Jmeter
- Oracle (previous Empirix)
- Quotium: Qtest
- Radview: WebLOAD
- WAPT
- Visual Studio 2010 Ultimate edition
- Apica Load Test
- Other, what?

6. Testing tool evaluation and licensies

6.1 Test management tools evaluation and licences

= A response to this question is required

Grade the tools you have chosen from 1 to 5. (1=unsatisfied, 5= excellent)

What kind of licences these selected tools have?

6.2 Incident management tools evaluation and licences

* A response to this question is required

Grade the tools you have chosen from 1 to 5. (1=unsatisfied, 5= excellent)

What kind of licences these selected tools have?

6.3 Static analysis tools evaluation and licences

* A response to this question is required

Grade the tools you have chosen from 1 to 5. (1=unsatisfied, 5= excellent)

What kind of licences these selected tools have?

6.4 Test data preparation tools evaluation and licences

* A response to this question is required

Grade the tools you have chosen from 1 to 5. (1=unsatisfied, 5= excellent)

What kind of licences these selected tools have?

6.5 Test execution tools evaluation and licences

* A response to this question is required

Grade the tools you have chosen from 1 to 5. (1=unsatisfied, 5= excellent)

What kind of licences these selected tools have?

6.6 Test design tools evaluation and licences

* A response to this question is required

Grade the tools you have chosen from 1 to 5. (1=unsatisfied, 5= excellent)

What kind of licences these selected tools have?

6.7 Test design tools evaluation and licences

* A response to this question is required

Grade the tools you have chosen from 1 to 5. (1=unsatisfied, 5= excellent)

What kind of licences these selected tools have?

6.8 Specialized unit test frameworks evaluation and licences

* A response to this question is required

Grade the tools you have chosen from 1 to 5. (1=unsatisfied, 5= excellent)

What kind of licences these selected tools have?

6.9 Dynamic analysis tools evaluation and licences

* A response to this question is required

Grade the tools you have chosen from 1 to 5. (1=unsatisfied, 5= excellent)

What kind of licences these selected tools have?

6.10 Performance/ load/ stress testing tools evaluation and licences

* A response to this question is required

Grade the tools you have chosen from 1 to 5. (1=unsatisfied, 5= excellent)

What kind of licences these selected tools have?

7. Use of testing tools

* A response to this question is required

7.1 Do you use office tools in testing (excel, word, wordpad, etc)?

- Yes
- No

7.2 If you selected ”yes”, what purpose do you use these tools?

7.3 To which functions do you use the testing tools?

- Reporting
- Test execution
- Planning test cases
- Generating test cases
- Requirement management
- Defect tracking
- Defect control and management
- Reviews
- Other function, what?

7.4 Are there enough testing tools in your company?

- Yes
- No

7.5 If you selected option “no”, what new tools are needed?

7.6 Are the current testing tool functionalities enough?

- Yes
- No

7.7 If you selected option “no”, what new functionalities are needed?

7.8 Are the current software testing tools used in all projects?

- Yes
- No

7.9 If you selected option ”no”, why not?

7.10 Are you utilizing current tools enough in testing?

- Yes
- No
- I don't know

7.11 If you selected option ”no”, why not?

7.12 Are some of current tools not fully utilized or are some of them not at all used because of some missing or bad functionality?

- Yes
- No
- I don't know

7.13 If you selected option ”yes”, what was the reason that testing tools are not used?

8. Education and empowerment

* A response to this question is required

8.1 What kind of training have you got for software testing?

- ISEB Practitioner
- ISEB Intermediate
- ISTQB/ISEB Foundation
- ISTQB Advanced level: Test Manager
- ISTQB Advanced level: Test Analyst
- ISTQB Advanced level: Technical Test Analyst
- ISTQB Expert level: Improving the Test Process
- ISTQB Expert level: Test Management
- Scrum master
- Tool-specific course
- Testing process course
- I have not had training
- Some other, what?

8.2 Is the training enough for your current software tools?

- Yes
- No

8.3 If you selected option ”no”, what kind of training would you need to have?

8.4 Does your employer/company intend to invest in new testing tools?

- Yes
- No
- I don’t know

8.5 If you selected option ”yes” what kind of tools will be looked for?

8.6 Have you participated in selecting software testing tool, or will you?

- Yes
- No
- I don't know

8.7 If your company will invest to new testing tools, who will select them?

- Testers
- Developers
- Business people
- Requirement engineers / business analysts
- Project manager
- Scrum master
- Software architect
- Software manager
- Test lead/ Test manager/ Test coordinator
- Outsourced testing professional
- Management of the company
- Somebody else, who?

8.8 Which functionalities will affect acquiring software testing tools?

- Open Source
- Management
- Ease of use
- Price
- Coverage
- Support for deployment
- Qualified documentation
- Better quality
- Visibility
- Changeability/ adaptability
- Speed
- Tool upgrades
- Tool features
- Reliability
- Transformability
- Reporting
- Releasing resources
- Follow up and measurements
- Performance
- Efficiency
- Tester’s choice
- Vendor’s characteristics (reputation, reliability, leader in the market etc)
- Repeatable
- Supported platforms
- Functionability
- Support in the future
- Getting less defects
- Compatible with other applications
- Commonness
- Product support (phone, online, email etc.)
- Training
- Return of investment (ROI)
- Cut costs in testing
- Save time in testing
- Reduce errors in applications
- I don’t know
- Some other, what?

9. Voluntary feedback

9.1 Your experiences/comments/ideas about selecting software testing tools for a company?

Thank you

Thank you for participating in our survey. Every reply is very valuable to us!

If you want to participate in the draw of Finnkino movie ticket package, leave your contact information, please. Survey will be done anonymously and confidentiality is guaranteed. Contact information cannot be combined to answers.

If you want to participate in the draw please fill your contact details below:
Name:

Address:

Email address:

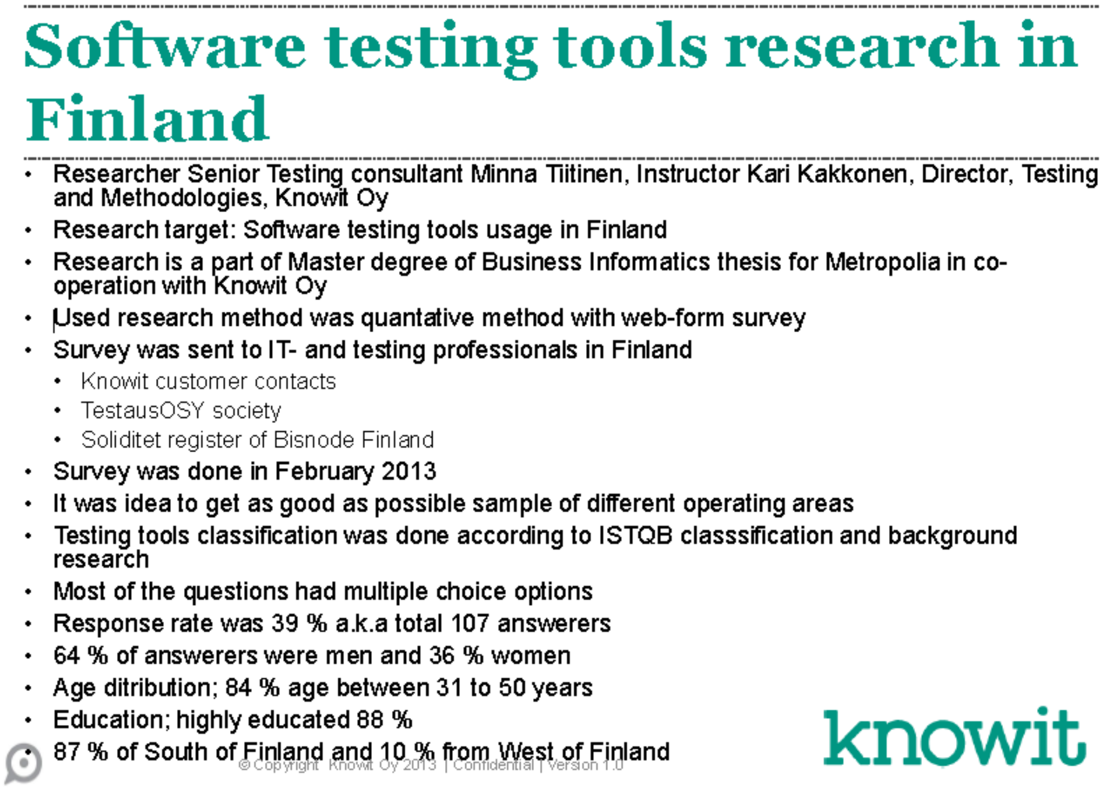
If you have any questions related to the survey, don't hesitate to contact to
minna.tiitinen@knowit.fi

The draw winner will receive personal email.

Appendix 2

Public report

Public report can be found also from the internet page of Knowit Oy, in the news; http://www.knowit.fi/Ajankohtaista/Pilvipalvelut-eivat-ole-viela-yleistyneet-testaustyokalujen-kaytossa/

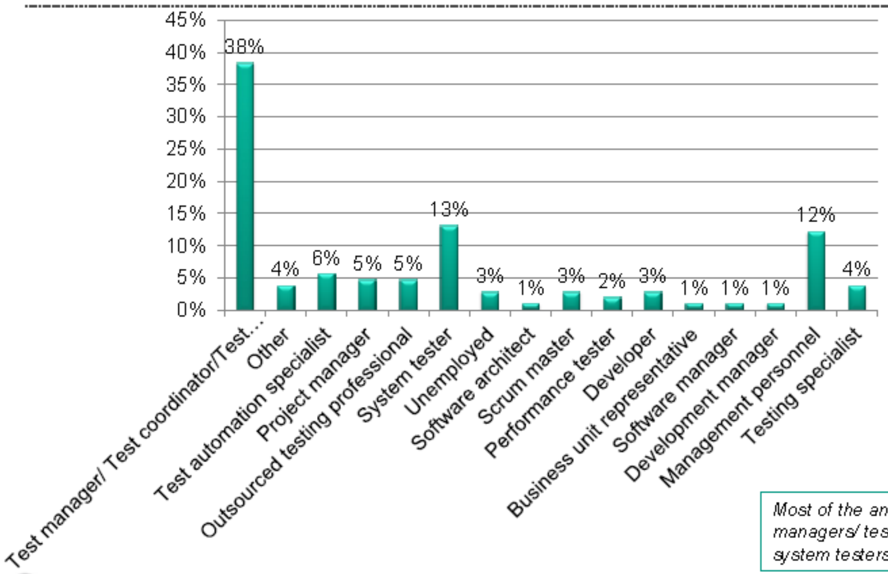


Software testing tools research in Finland

- Researcher Senior Testing consultant Minna Tiitinen, Instructor Kari Kakkonen, Director, Testing and Methodologies, Knowit Oy
- Research target: Software testing tools usage in Finland
- Research is a part of Master degree of Business Informatics thesis for Metropolia in co-operation with Knowit Oy
- Used research method was quantative method with web-form survey
- Survey was sent to IT- and testing professionals in Finland
  - Knowit customer contacts
  - TestausOSY society
  - Soliditet register of Bisnode Finland
- Survey was done in February 2013
- It was idea to get as good as possible sample of different operating areas
- Testing tools classification was done according to ISTQB classsification and background research
- Most of the questions had multiple choice options
- Response rate was 39 % a.k.a total 107 answerers
- 64 % of answerers were men and 36 % women
- Age ditribution; 84 % age between 31 to 50 years
- Education; highly educated 88 %
- 87 % of South of Finland and 10 % from West of Finland

© Copyright Knowit Oy 2013 | Confidential | Version 1.0

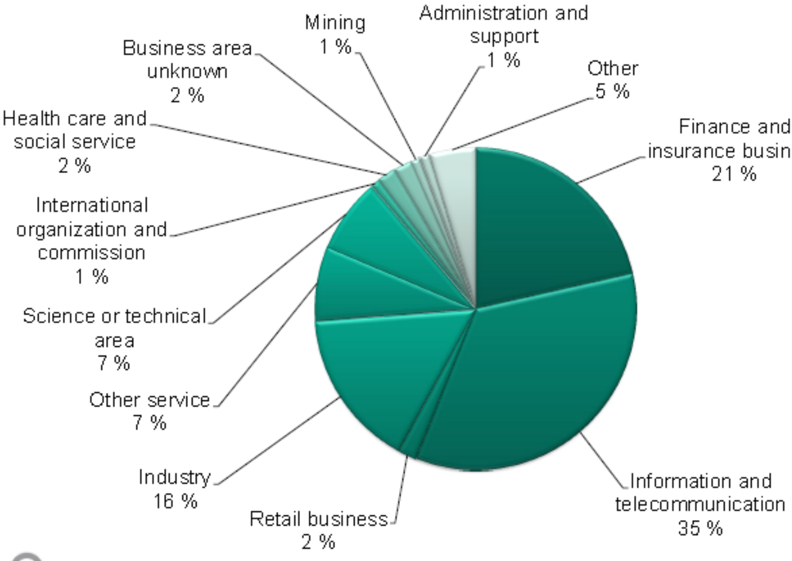
Profession/role in the company



Most of the answerers were test managers/ test coordinators and system testers

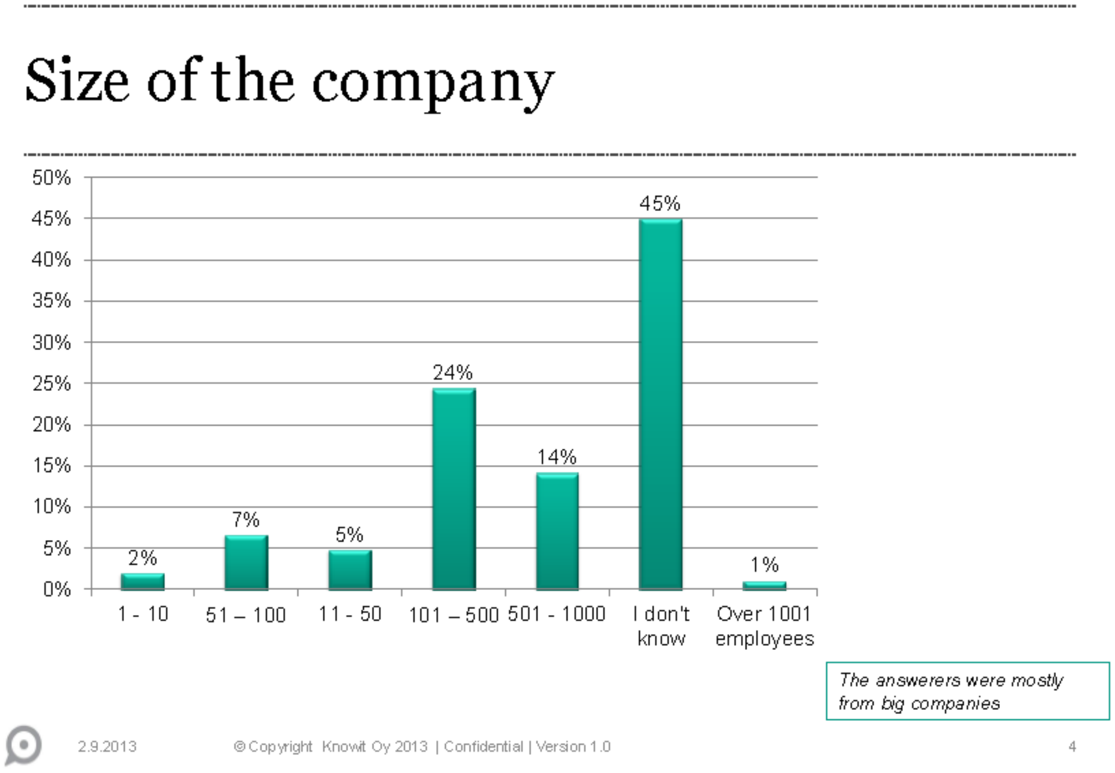
2.9.2013 © Copyright Knowit Oy 2013 | Confidential | Version 1.0 2

Operating area of the company



The biggest operating areas were Information and telecommunication, Finance and insurance business and Industry

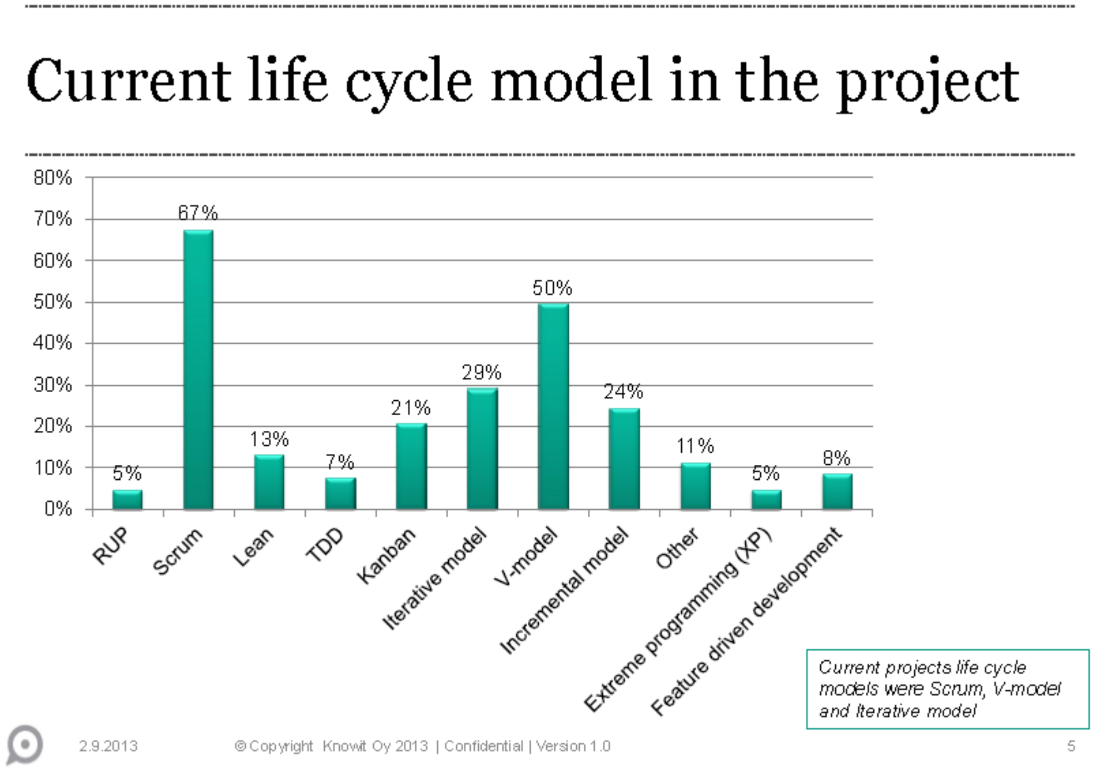
2.9.2013 © Copyright Knowit Oy 2013 | Confidential | Version 1.0 3



Size of the company

The answerers were mostly from big companies

2.9.2013 ©Copyright Knowit Oy 2013 | Confidential | Version 1.0 4

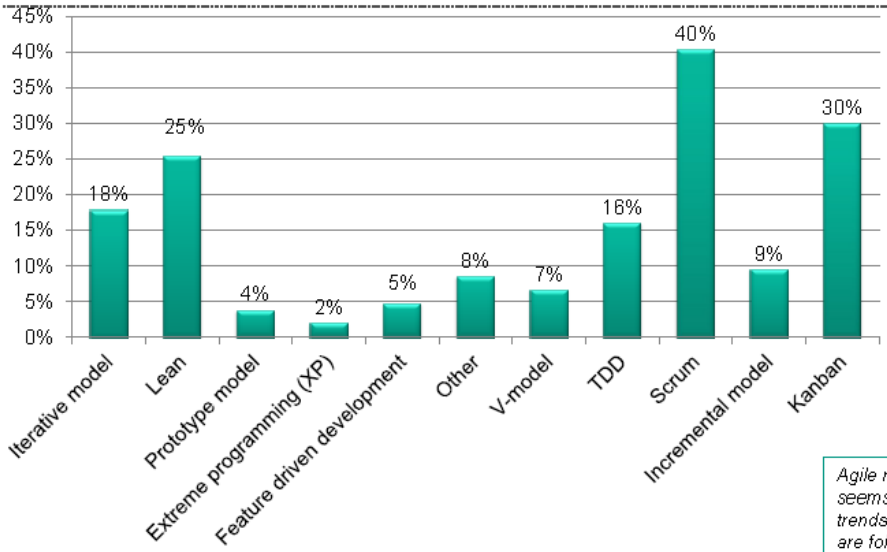


Current life cycle model in the project

Current projects life cycle models were Scrum, V-model and Iterative model

2.9.2013 ©Copyright Knowit Oy 2013 | Confidential | Version 1.0 5

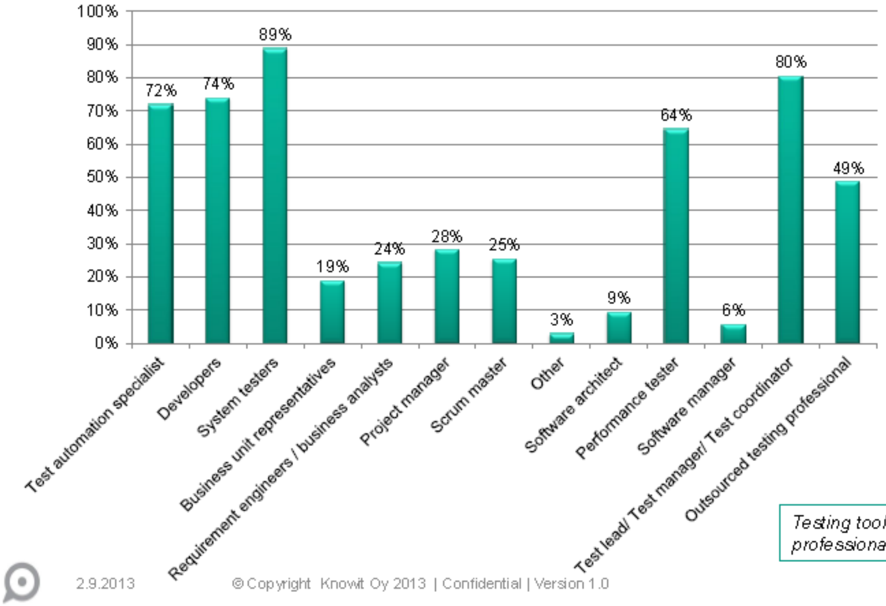
Future life cycle model



Agile methods like scrum seems to be future trends, Kanban and Lean are following, V-model use is going down

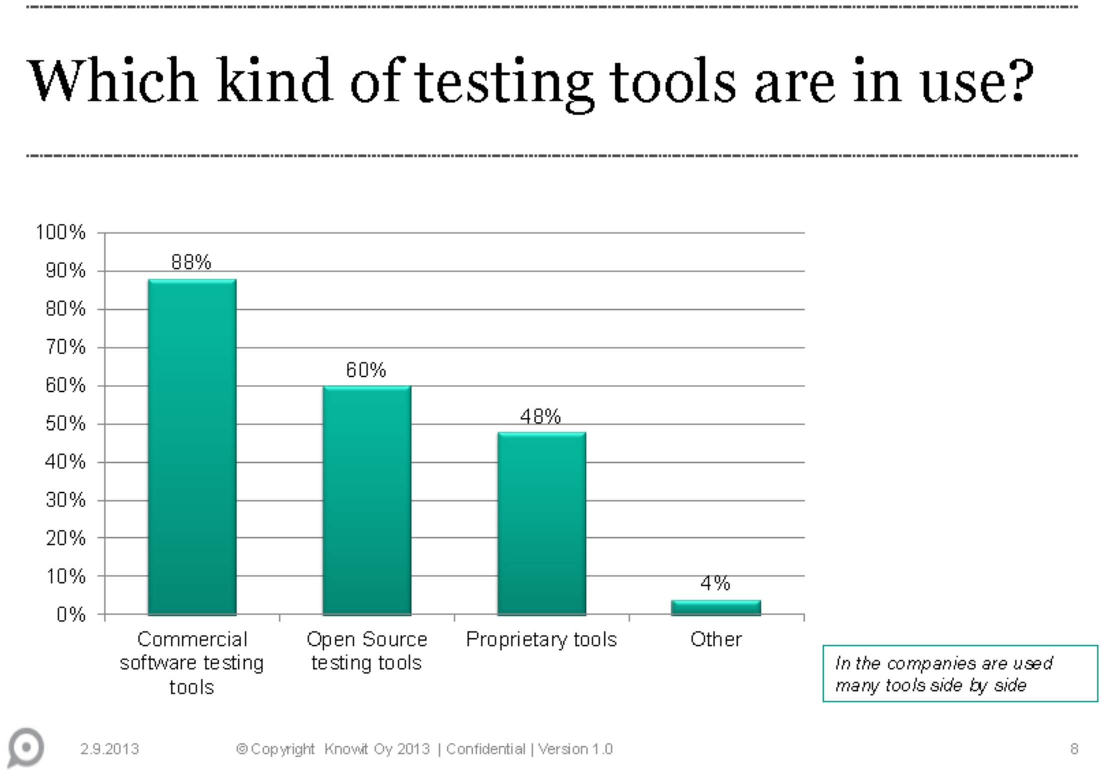
2.9.2013 © Copyright Knowit Oy 2013 | Confidential | Version 1.0 6

Which employees of your company are using software testing tools?



Testing tools are used by testing professionals and developers

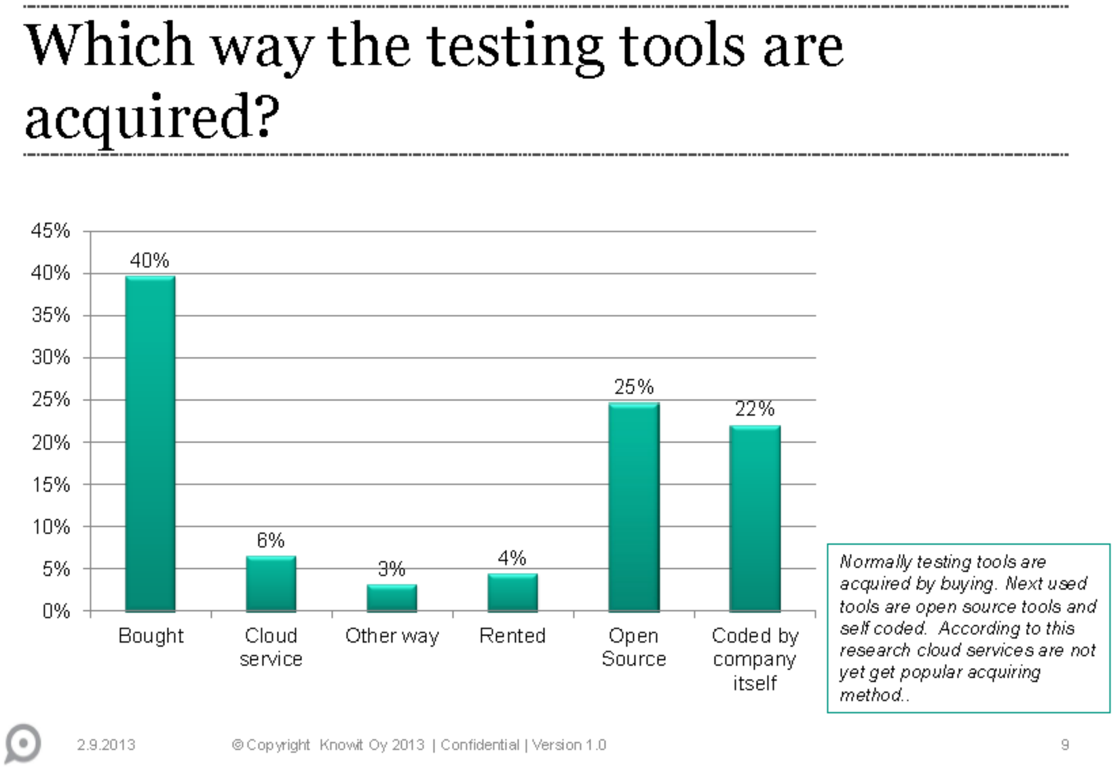
2.9.2013 © Copyright Knowit Oy 2013 | Confidential | Version 1.0 7



Which kind of testing tools are in use?

In the companies are used many tools side by side

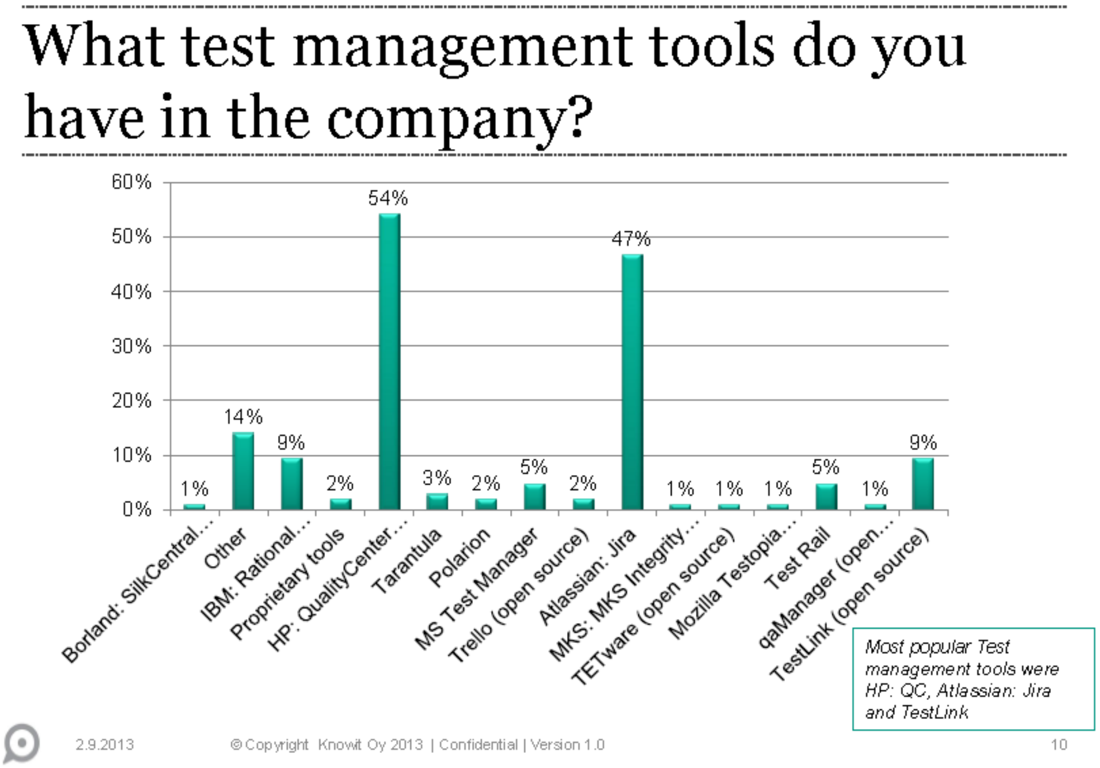
2.9.2013 © Copyright Knowit Oy 2013 | Confidential | Version 1.0 8



Which way the testing tools are acquired?

Normally testing tools are acquired by buying. Next used tools are open source tools and self coded. According to this research cloud services are not yet get popular acquiring method..

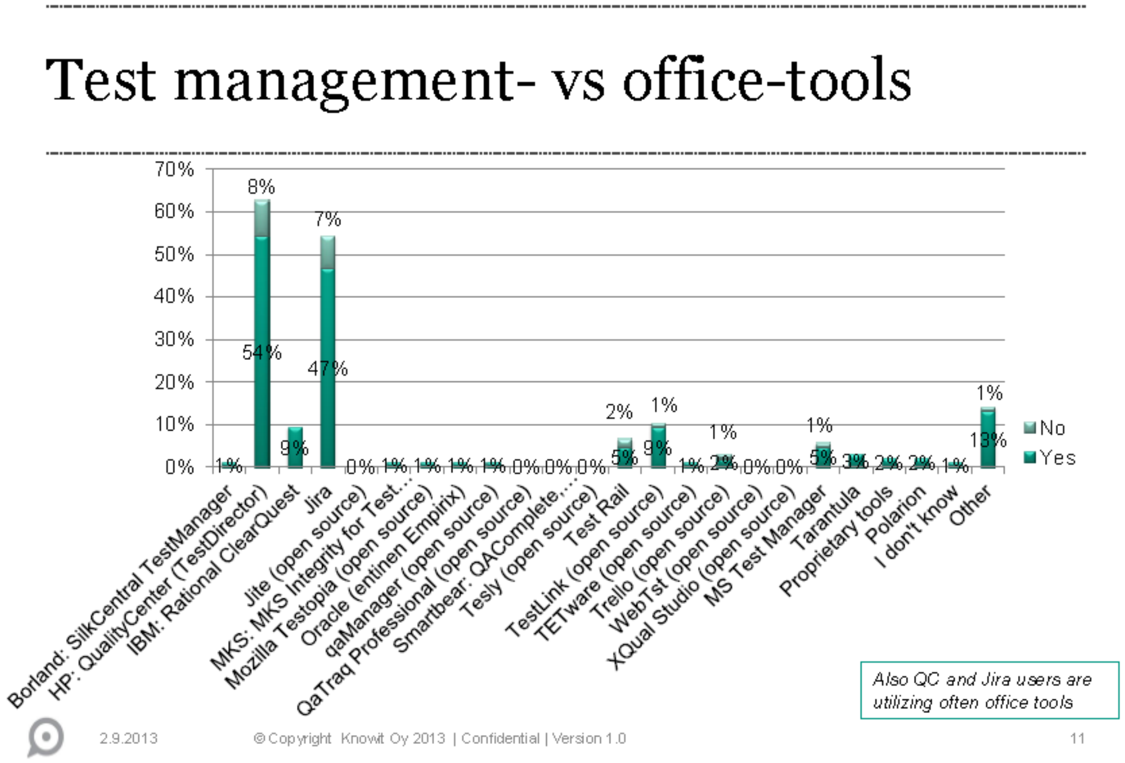
2.9.2013 © Copyright Knowit Oy 2013 | Confidential | Version 1.0 9



What test management tools do you have in the company?

Most popular Test management tools were HP: QC, Atlassian: Jira and TestLink

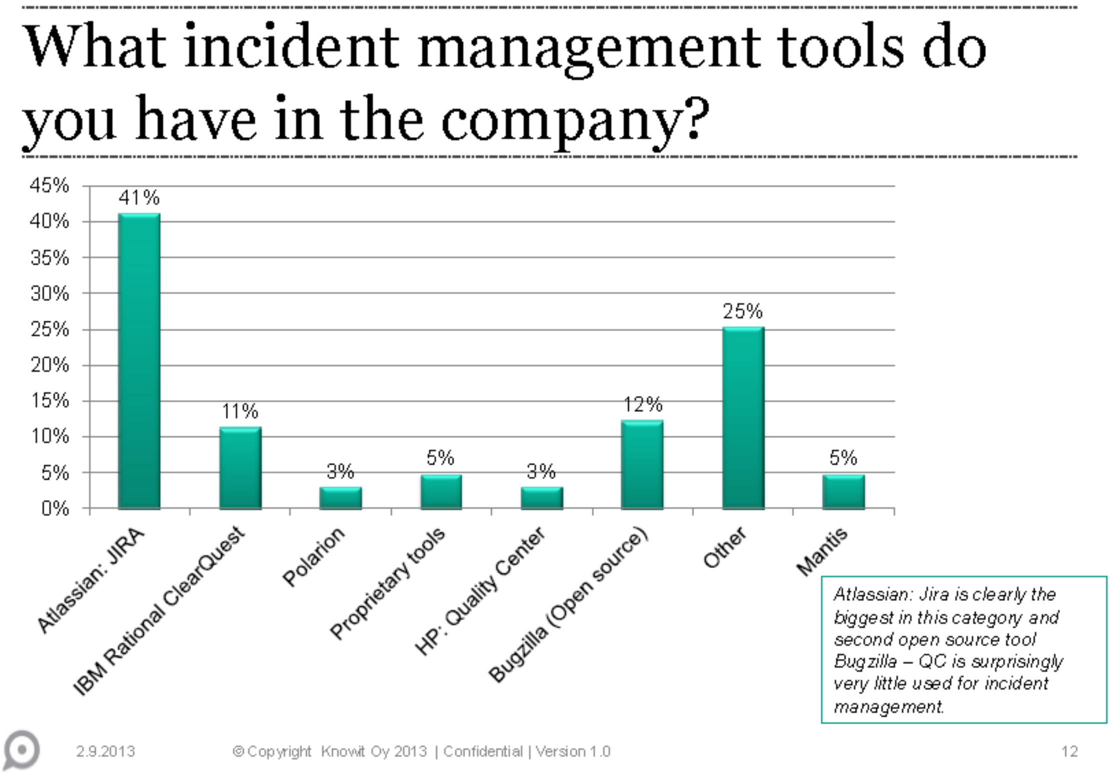
2.9.2013 © Copyright Knowit Oy 2013 | Confidential | Version 1.0 10



Test management- vs office-tools

Also QC and Jira users are utilizing often office tools

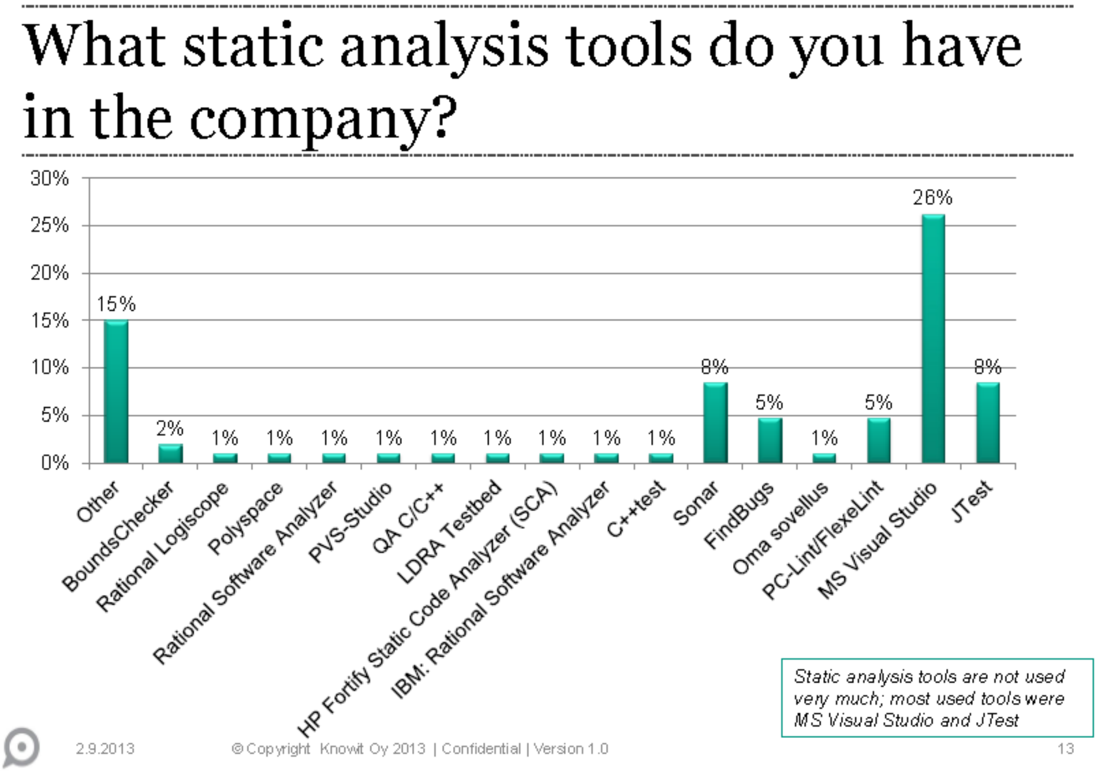
2.9.2013 © Copyright Knowit Oy 2013 | Confidential | Version 1.0 11



What incident management tools do you have in the company?

Atlassian: Jira is clearly the biggest in this category and second open source tool Bugzilla – QC is surprisingly very little used for incident management.

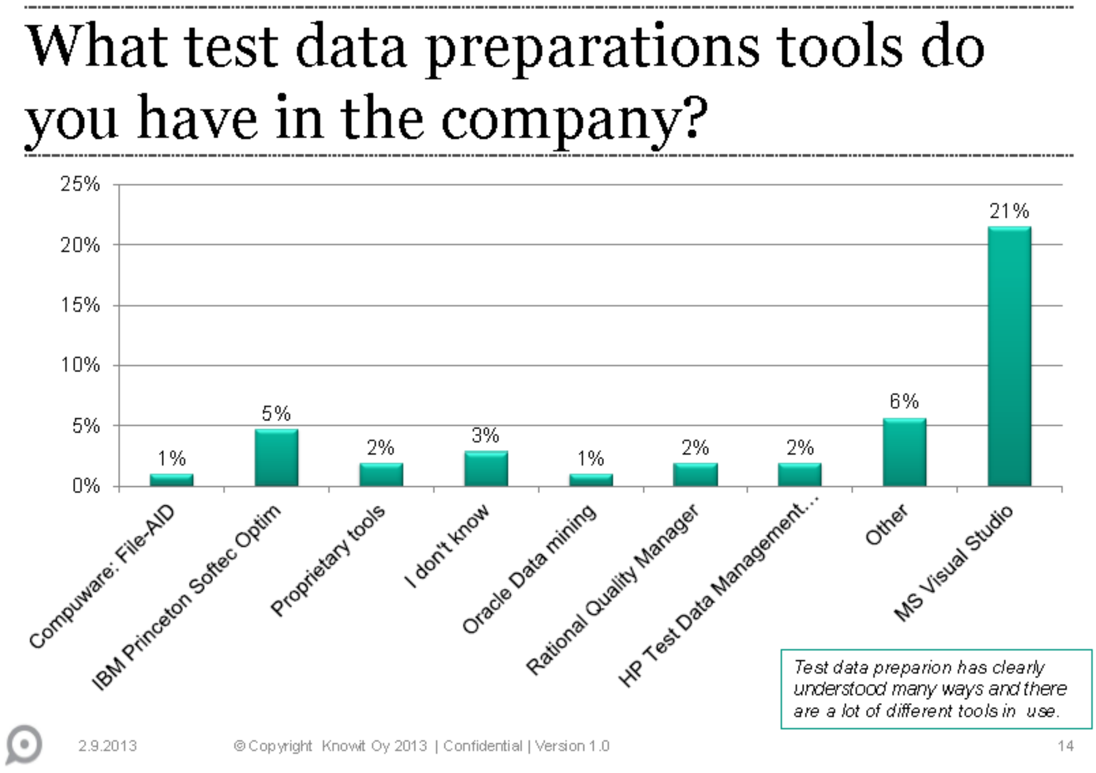
2.9.2013 © Copyright Knowit Oy 2013 | Confidential | Version 1.0 12



What static analysis tools do you have in the company?

Static analysis tools are not used very much; most used tools were MS Visual Studio and JTest

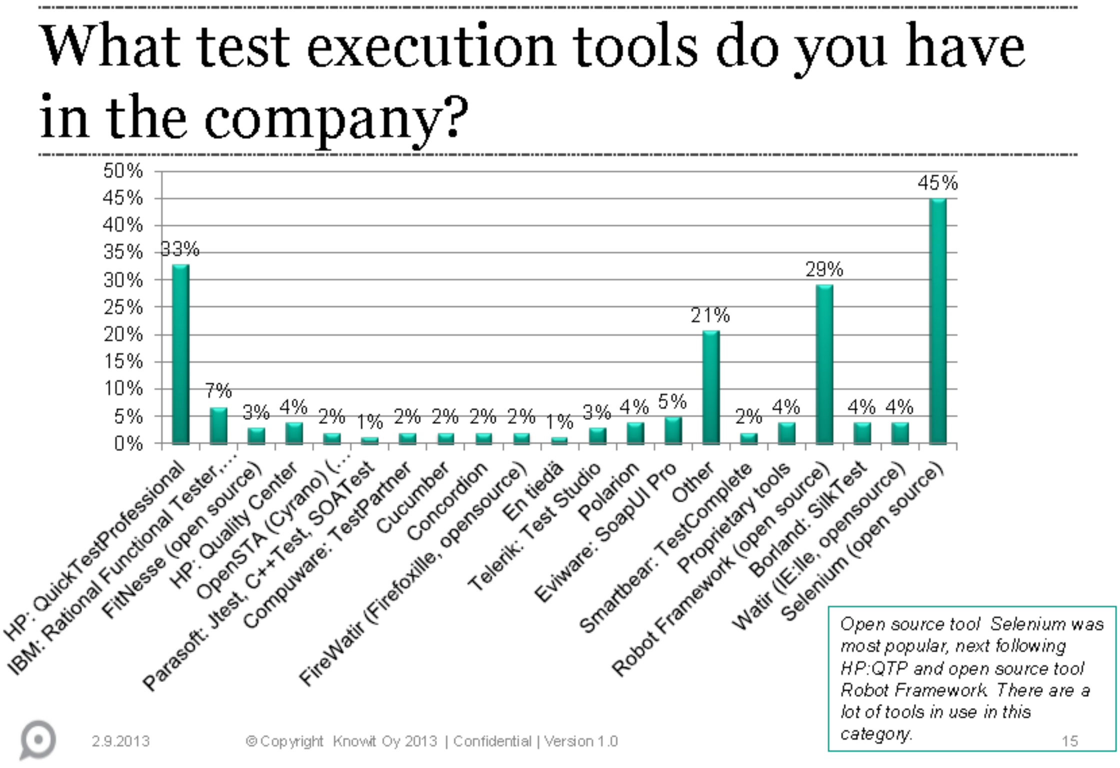
2.9.2013 © Copyright Knowit Oy 2013 | Confidential | Version 1.0 13



What test data preparations tools do you have in the company?

Test data preparion has clearly understood many ways and there are a lot of different tools in use.

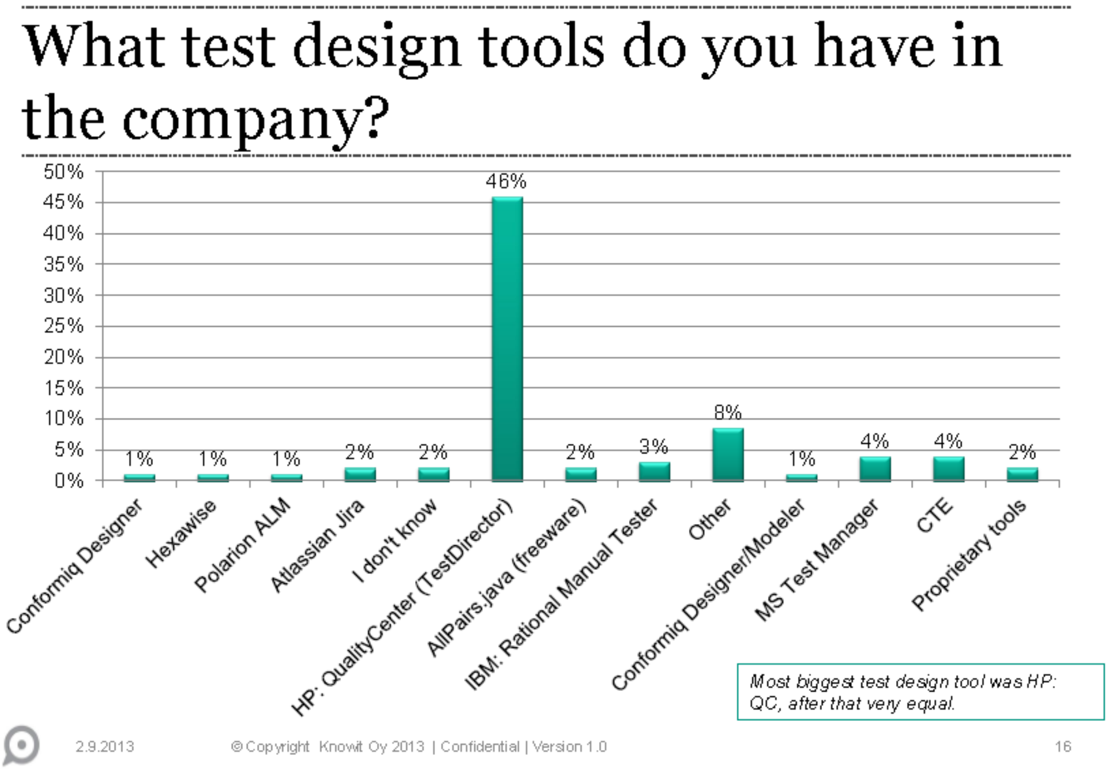
2.9.2013 © Copyright Knowit Oy 2013 | Confidential | Version 1.0 14



What test execution tools do you have in the company?

Open source tool Selenium was most popular, next following HP:QTP and open source tool Robot Framework. There are a lot of tools in use in this category.

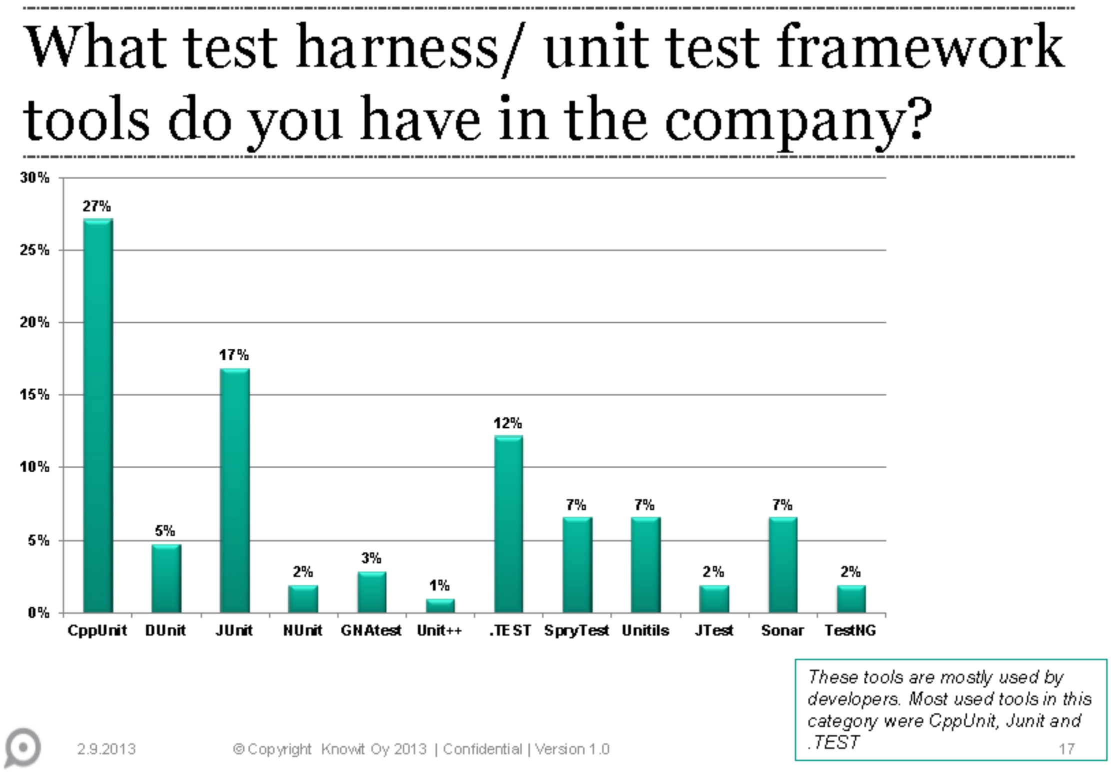
2.9.2013 © Copyright Knowit Oy 2013 | Confidential | Version 1.0 15



What test design tools do you have in the company?

Most biggest test design tool was HP: QC, after that very equal.

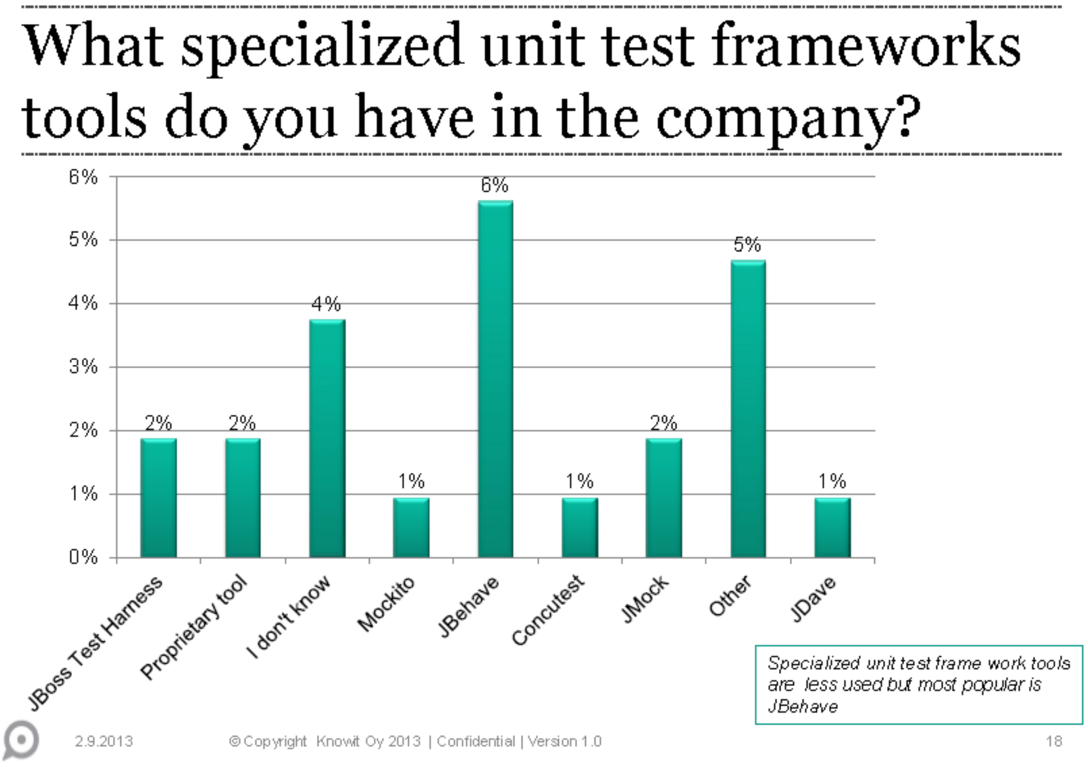
2.9.2013 © Copyright Knowit Oy 2013 | Confidential | Version 1.0 16



What test harness/ unit test framework tools do you have in the company?

These tools are mostly used by developers. Most used tools in this category were CppUnit, Junit and .TEST

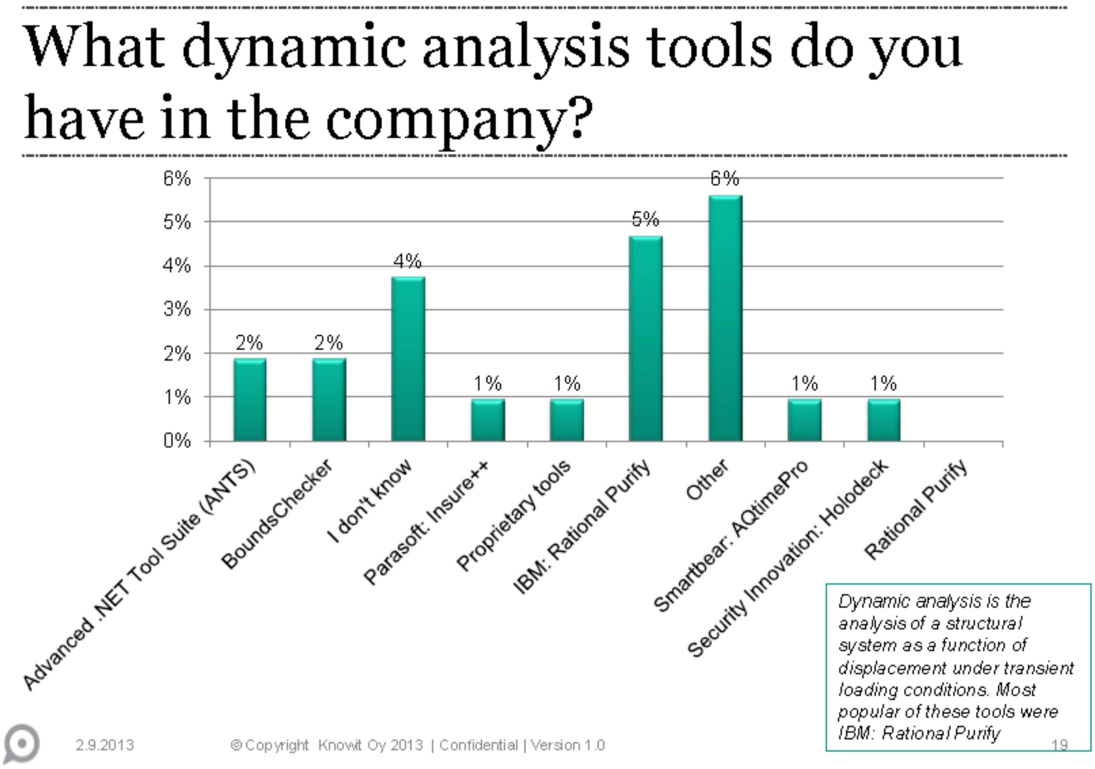
2.9.2013 © Copyright Knowit Oy 2013 | Confidential | Version 1.0 17



What specialized unit test frameworks tools do you have in the company?

Specialized unit test frame work tools are less used but most popular is JBehave

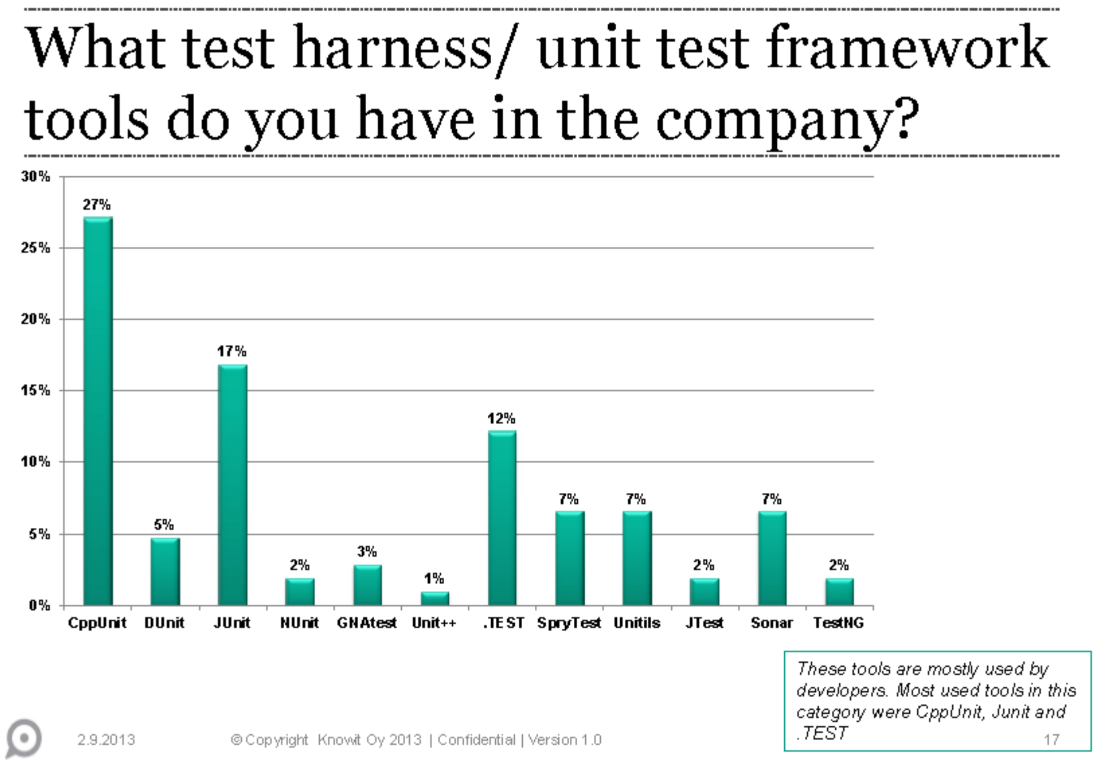
2.9.2013 ©Copyright Knowit Oy 2013 | Confidential | Version 1.0 18



What dynamic analysis tools do you have in the company?

Dynamic analysis is the analysis of a structural system as a function of displacement under transient loading conditions. Most popular of these tools were IBM: Rational Purify

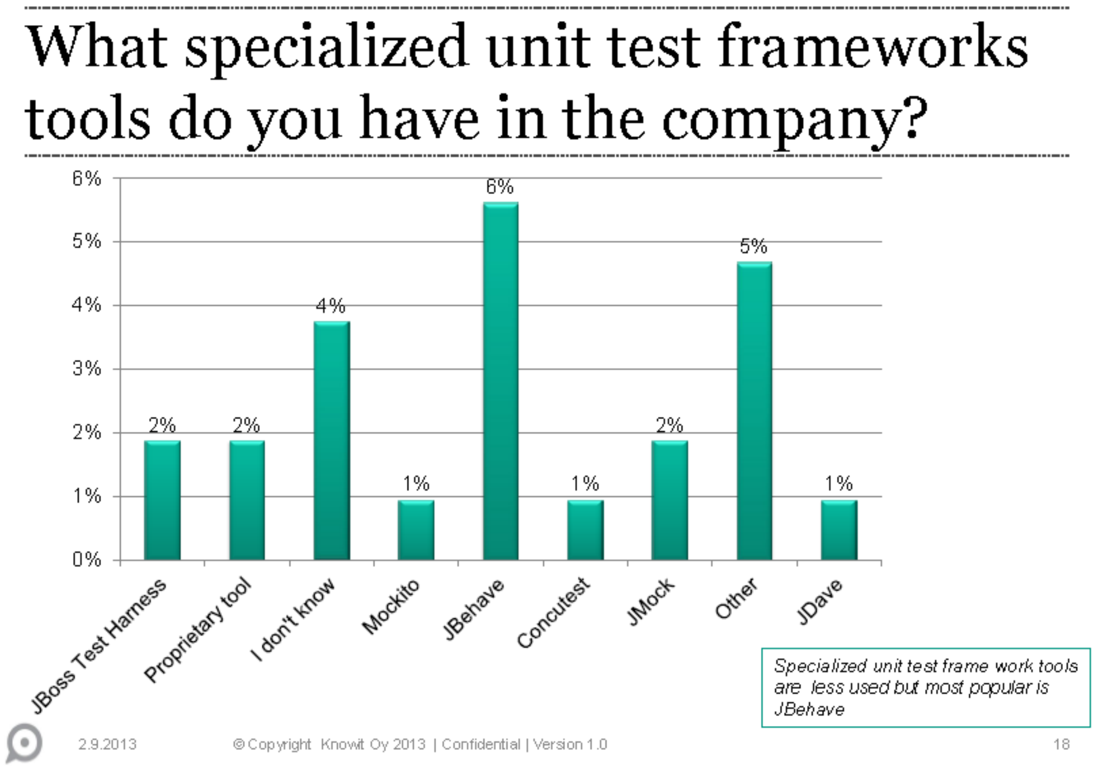
2.9.2013 ©Copyright Knowit Oy 2013 | Confidential | Version 1.0 19



What test harness/ unit test framework tools do you have in the company?

These tools are mostly used by developers. Most used tools in this category were CppUnit, Junit and .TEST

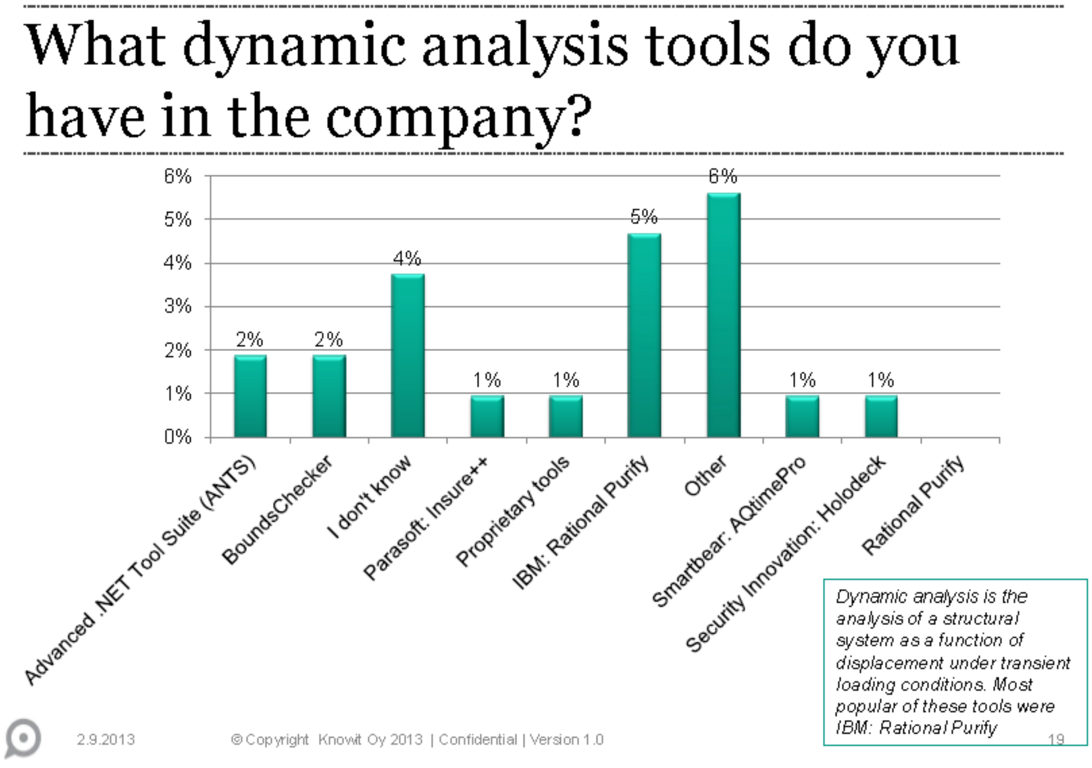
2.9.2013 ©Copyright Knowit Oy 2013 | Confidential | Version 1.0 17



What specialized unit test frameworks tools do you have in the company?

Specialized unit test frame work tools are less used but most popular is JBehave

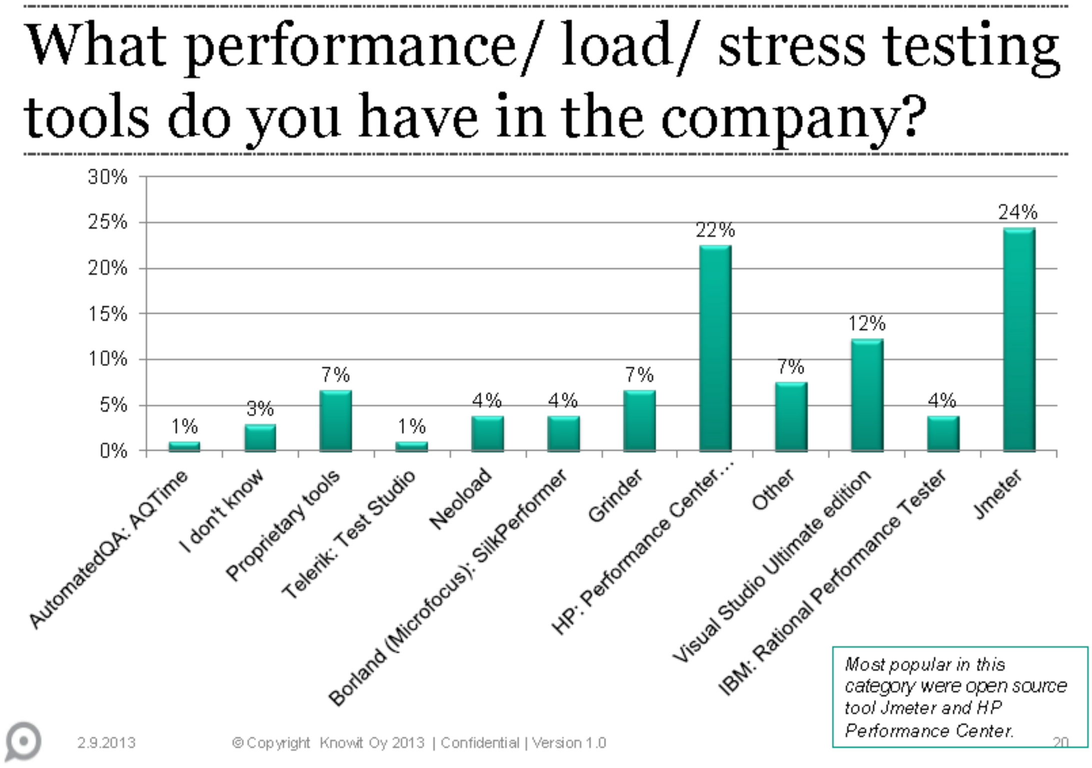
2.9.2013 ©Copyright Knowit Oy 2013 | Confidential | Version 1.0 18



What dynamic analysis tools do you have in the company?

Dynamic analysis is the analysis of a structural system as a function of displacement under transient loading conditions. Most popular of these tools were IBM: Rational Purify

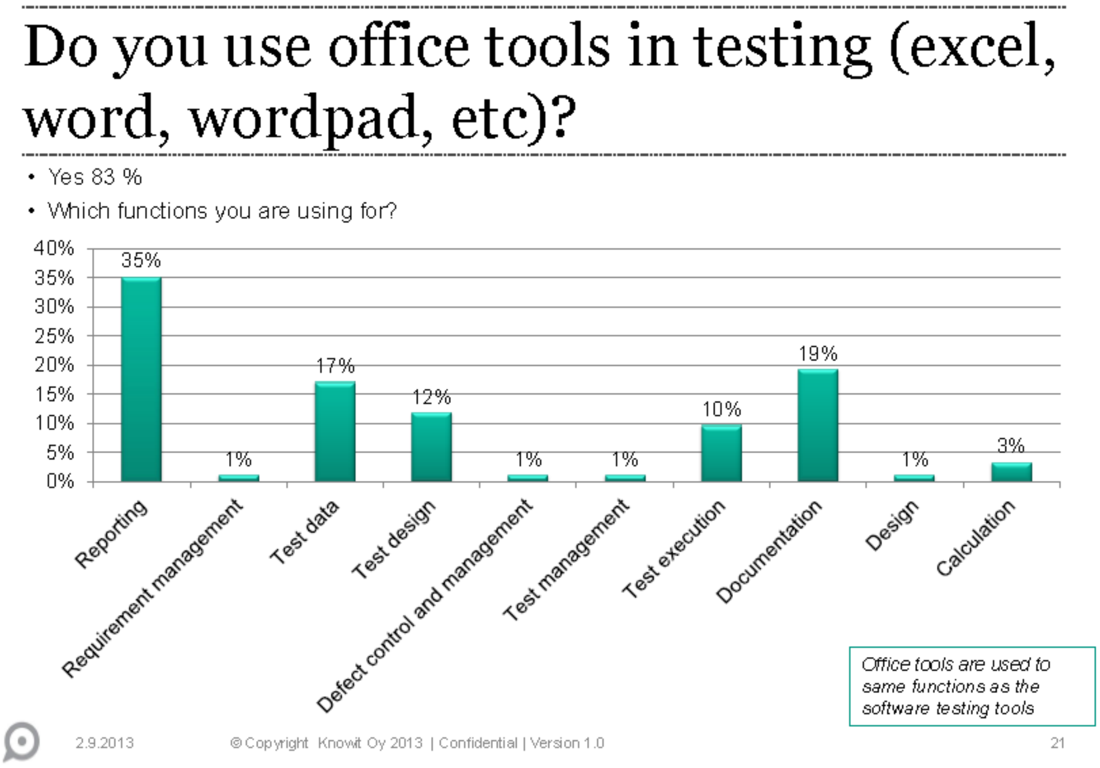
2.9.2013 © Copyright Knowit Oy 2013 | Confidential | Version 1.0 19



What performance/ load/ stress testing tools do you have in the company?

Most popular in this category were open source tool Jmeter and HP Performance Center.

2.9.2013 © Copyright Knowit Oy 2013 | Confidential | Version 1.0 20

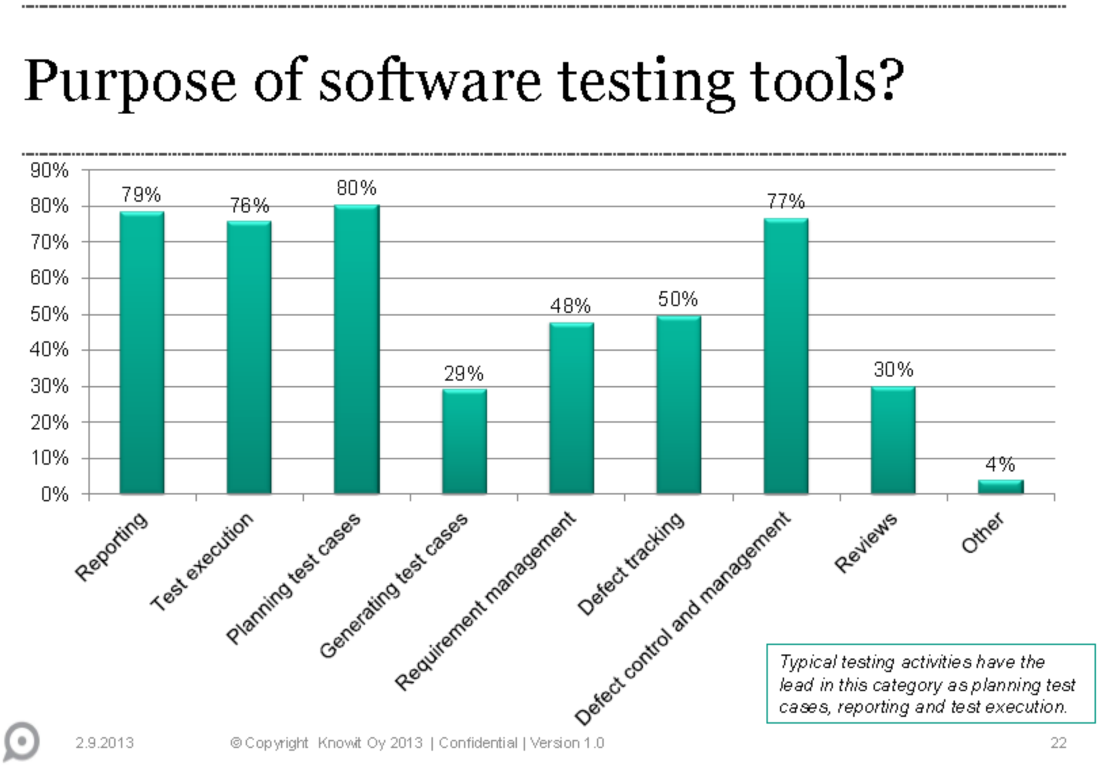


Do you use office tools in testing (excel, word, wordpad, etc)?

- Yes 83 %
- Which functions you are using for?

Office tools are used to same functions as the software testing tools

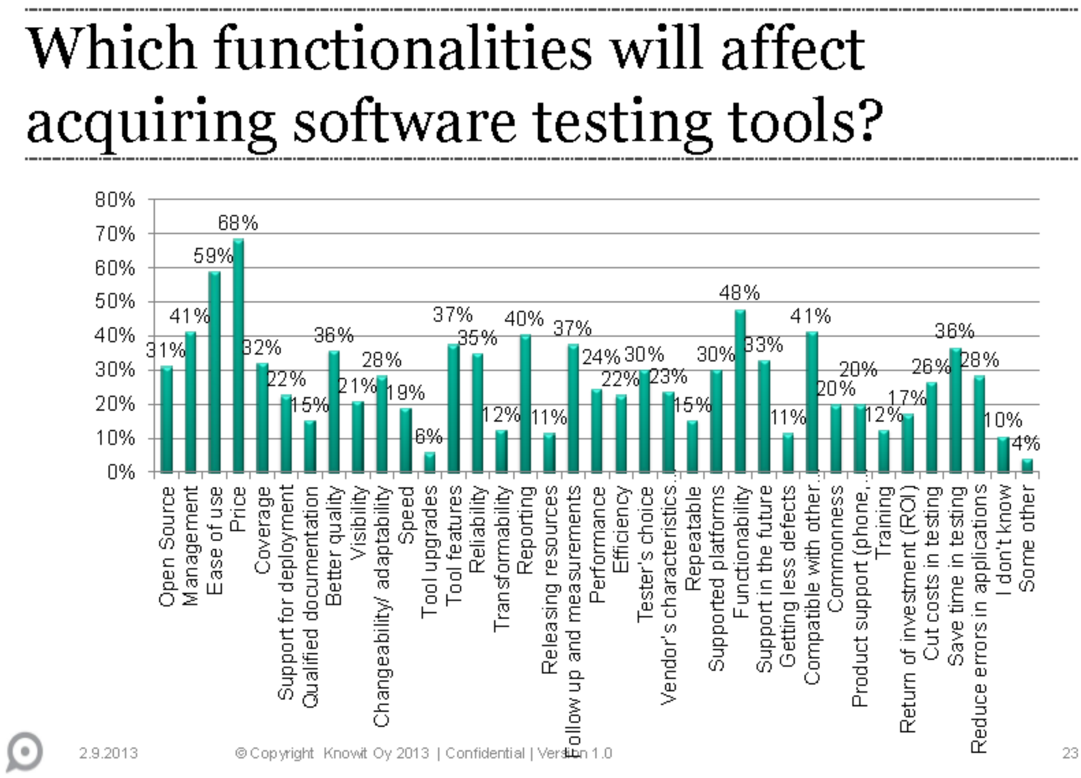
2.9.2013 © Copyright Knowit Oy 2013 | Confidential | Version 1.0 21



Purpose of software testing tools?

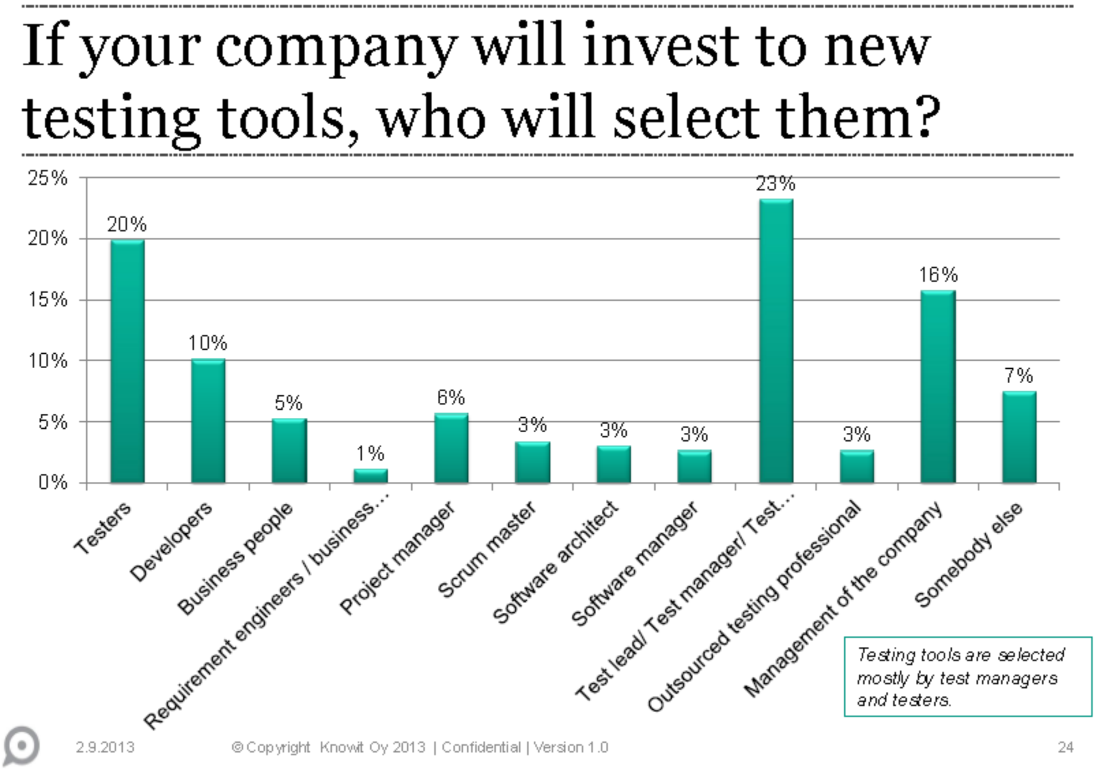
Typical testing activities have the lead in this category as planning test cases, reporting and test execution.

2.9.2013 © Copyright Knowit Oy 2013 | Confidential | Version 1.0 22



Which functionalities will affect acquiring software testing tools?

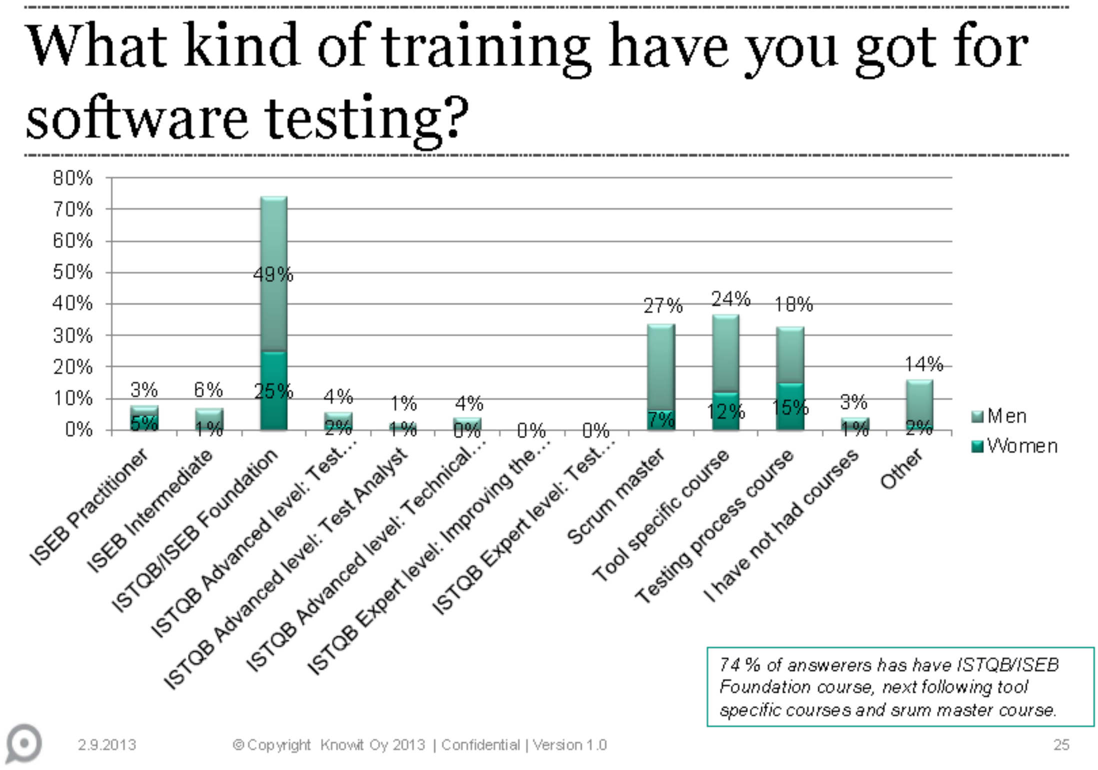
2.9.2013 © Copyright Knowit Oy 2013 | Confidential | Version 1.0 23



If your company will invest to new testing tools, who will select them?

Testing tools are selected mostly by test managers and testers.

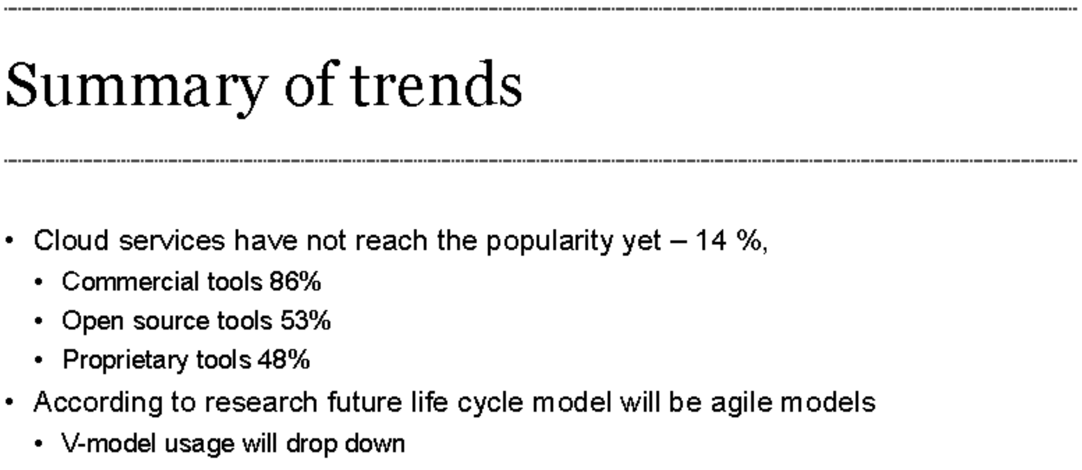
2.9.2013 © Copyright Knowit Oy 2013 | Confidential | Version 1.0 24



What kind of training have you got for software testing?

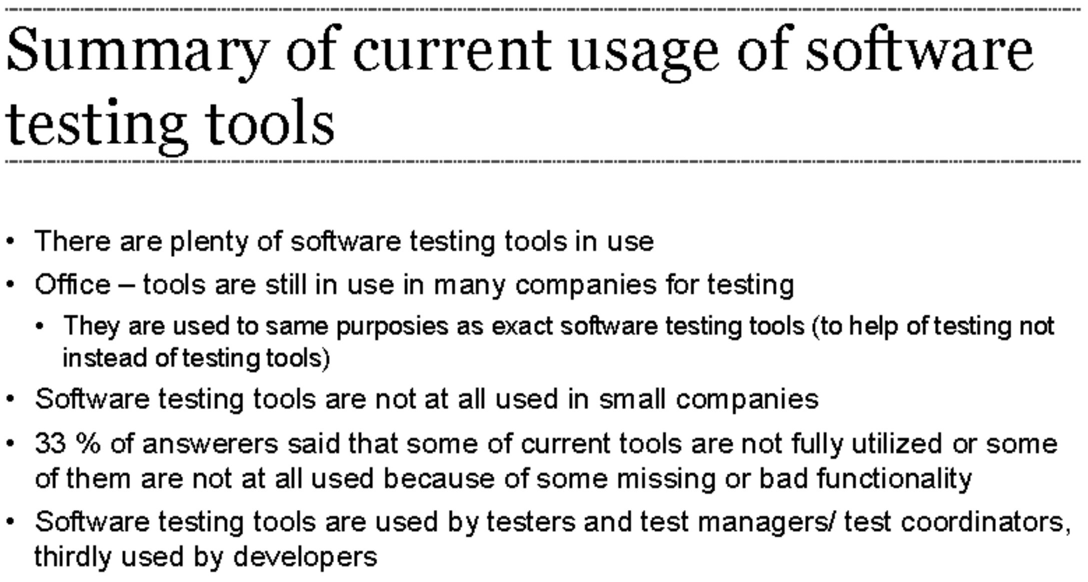
74 % of answerers has have ISTQB/ISEB Foundation course, next following tool specific courses and srum master course.

2.9.2013 ©Copyright Knowit Oy 2013 | Confidential | Version 1.0 25



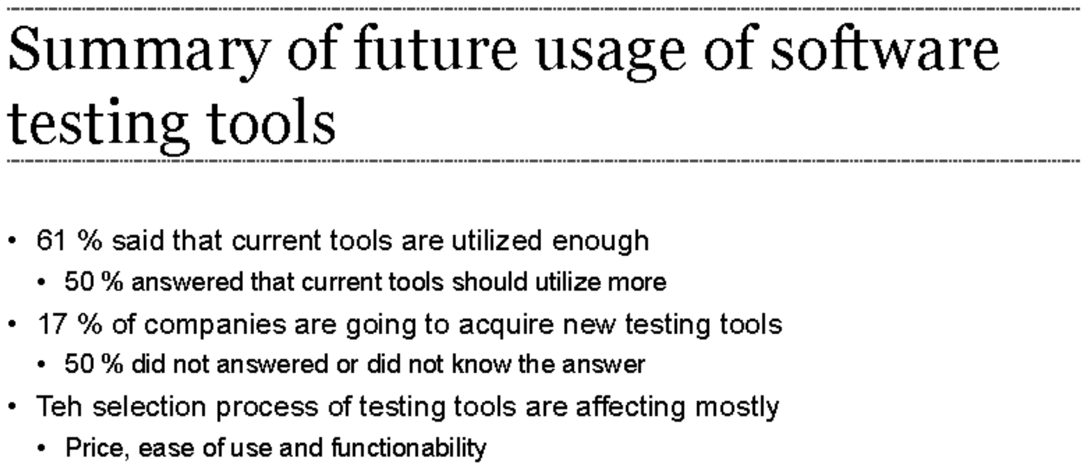
Summary of trends

- Cloud services have not reach the popularity yet – 14 %,
  - Commercial tools 86%
  - Open source tools 53%
  - Proprietary tools 48%
- According to research future life cycle model will be agile models
  - V-model usage will drop down



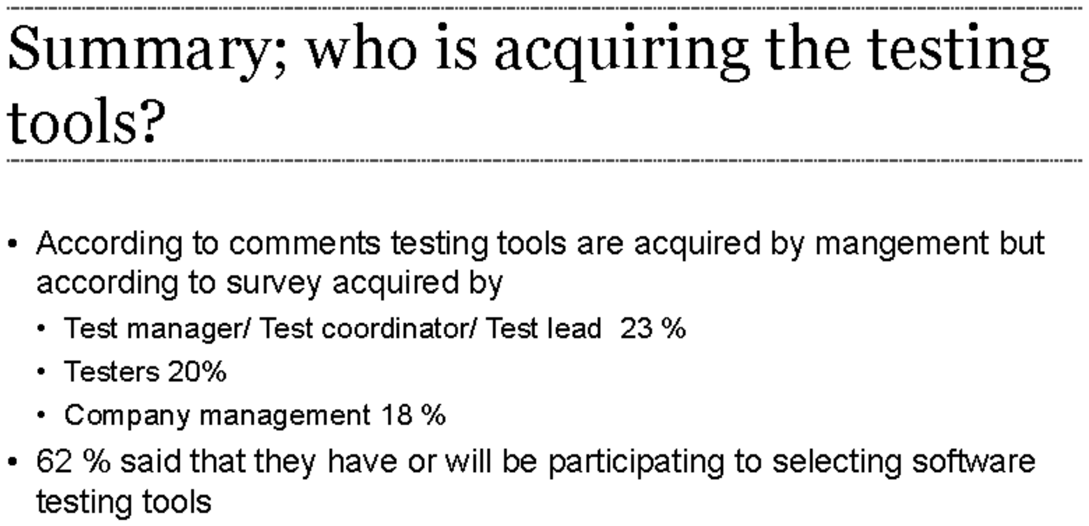
Summary of current usage of software testing tools

- There are plenty of software testing tools in use
- Office – tools are still in use in many companies for testing
  - They are used to same purposies as exact software testing tools (to help of testing not instead of testing tools)
- Software testing tools are not at all used in small companies
- 33 % of answerers said that some of current tools are not fully utilized or some of them are not at all used because of some missing or bad functionality
- Software testing tools are used by testers and test managers/ test coordinators, thirdly used by developers



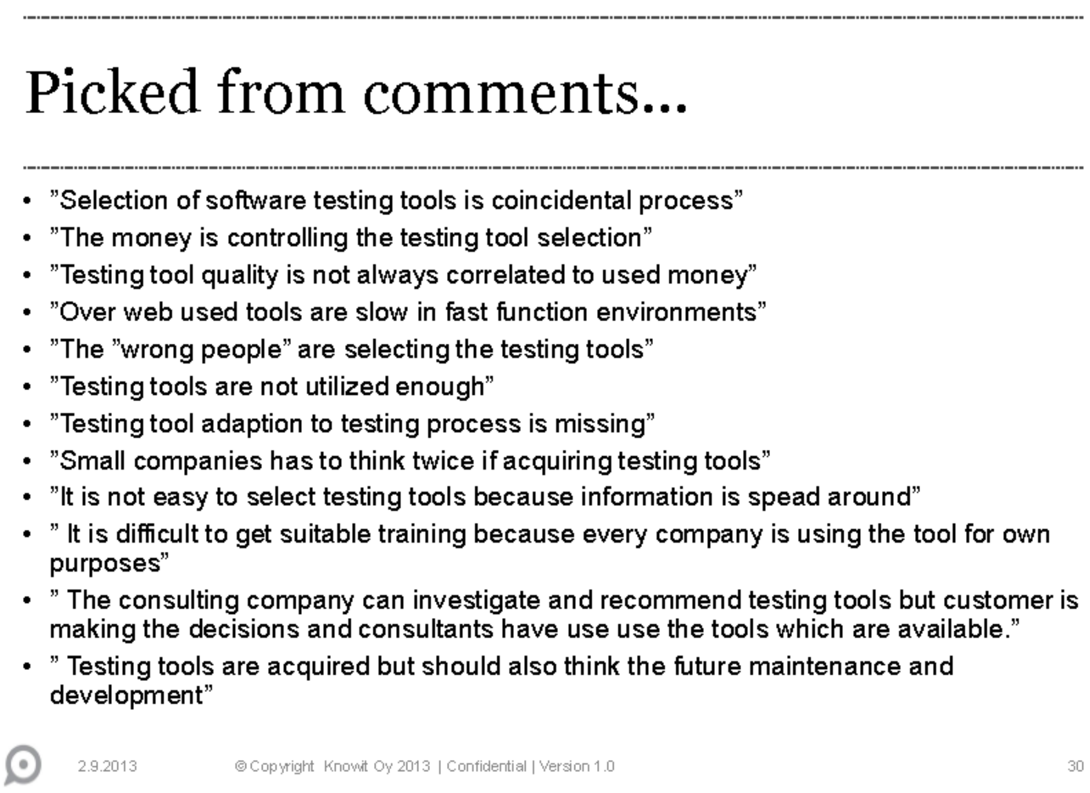
Summary of future usage of software testing tools

- 61 % said that current tools are utilized enough
  - 50 % answered that current tools should utilize more
- 17 % of companies are going to acquire new testing tools
  - 50 % did not answered or did not know the answer
- Teh selection process of testing tools are affecting mostly
  - Price, ease of use and functionability



Summary; who is acquiring the testing tools?

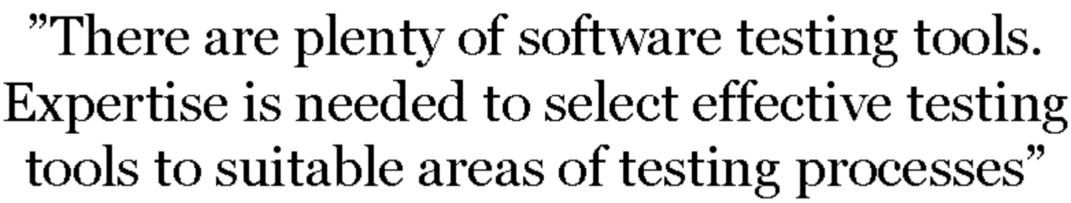
- According to comments testing tools are acquired by mangement but according to survey acquired by
  - Test manager/ Test coordinator/ Test lead 23 %
  - Testers 20%
  - Company management 18 %
- 62 % said that they have or will be participating to selecting software testing tools



Picked from comments...

- ”Selection of software testing tools is coincidental process”
- ”The money is controlling the testing tool selection”
- ”Testing tool quality is not always correlated to used money”
- ”Over web used tools are slow in fast function environments”
- ”The ”wrong people” are selecting the testing tools”
- ”Testing tools are not utilized enough”
- ”Testing tool adaption to testing process is missing”
- ”Small companies has to think twice if acquiring testing tools”
- ”It is not easy to select testing tools because information is spead around”
- ” It is difficult to get suitable training because every company is using the tool for own purposes”
- ” The consulting company can investigate and recommend testing tools but customer is making the decisions and consultants have use use the tools which are available.”
- ” Testing tools are acquired but should also think the future maintenance and development”

2.9.2013 © Copyright Knowit Oy 2013 | Confidential | Version 1.0 30



”There are plenty of software testing tools. Expertise is needed to select effective testing tools to suitable areas of testing processes”

Appendix 3

Testing tool selection list (Lewis, W.E., 2004)

Sr. no	Check Point / Defect Statement		
		Yes	N/a
1	Whether the tool is easy for your testers to use? Is it something that can be picked up quickly, or is training going to be required.		
2	Whether any of the team members already have experience of using the tool.		
3	Whether training is necessary? Whether classes, books, or other forms of instruction are available.		
4	Whether the tool will work effectively on the computer system currently in place.		
5	Whether more memory, faster processors, etc., going to be needed.		
6	Whether the tool itself is easy to use.		
7	Whether the tool has a user-friendly interface.		
8	Whether the tool is prone to user error.		
9	Whether the tool is physically capable of testing your application? It may be noted that many testing tools can only test in a GUI environment, while others test in non-GUI environments.		
10	Whether the tool can handle full project testing? That is, is it able to run hundreds if not thousands of test cases for extended periods of time.		
11	Whether the tool can run for long periods of time without crashing, or is the tool itself full of bugs.		
12	Whether you have talked to the customers who currently or previously have used the tool. Did it meet		

	their needs.		
13	How similar were their testing needs to yours and how well did the tool perform.		
14	Whether you are trying to select a tool that is advanced enough so the costs of updating tests don’t overwhelm any benefits of testing.		
15	Whether a demo version of the tool is available? Have you tried it out before making any decision.		
16	Whether the price of the tool fit in the QA department or company budget.		
17	Whether the tool meets the requirements of the company testing methodology.		