NTNU

Norwegian University of
Science and Technology

# Investigation of the use of test automation in software quality assurance in Norwegian companies and organisations

**Saraf Anika**

Master in Information Systems
Submission date: July 2016
Supervisor: Carl-Fredrik Sørensen, IDI

Norwegian University of Science and Technology
Department of Computer and Information Science

Saraf Anika

# Investigation of the use of Test Automation in Software Quality Assurance in Norwegian Companies and Organisations

Trondheim, July 2016

Master´s thesis

NTNU
Norwegian University of
Science and Technology
Faculty of Information Technology, Mathematics and
Electrical Engineering
Department of Computer and Information Science

NTNU
Norwegian University of
Science and Technology

NTNU

Norwegian University of
Science and Technology

Saraf Anika

# Investigation of the use of Test Atomation in Software Quality Assurance in Norwegian Companies and Organisations

Supervised By:
Carl-Fredrik Sørensen, Senior Lecturer

Trondheim, July 4 2016

Norwegian University of Science and Technology
Faculty of Information Technology, Mathematics and Electrical Engineering
Department of Computer and Information Science

NTNU

# Abstract

The focus of this thesis has been about automated testing in practice. Using testing tools is quite common phenomenon in software companies. A questionnaire about the use of testing tools in software quality assurance to investigate Norwegian software companies has been made. The questionnaire has been sent out to a number of Norwegian IT/software and non-IT/software companies to get feedback on their use of testing tools for projects and product development. Four interviews have been conducted to find out more about the real life practice of using testing tools in different companies. Quantitative Analysis has been used on the responses from the questionnaire and each response from the questionnaire have been statistically analyzed. The statistical analysis of the responses shows the percentage of how many people are using testing tools and why, who are still solely dependent on manual testing and why and how the software quality assurance testings is done. The main objective is to find out facts related to the research questions; the less practice of using testing tools in real life despite of its popularity in literature reviews and how the companies are following agile method and debugging at the same time.

## Acknowledgment

I want to thank all the people who are associated with this work.

I would like to express my gratitude to my supervisor, Carl-Fredrik Sørensen for his advice, help and suggestions regarding this project. His guidance for the project structure and extensive help on the questionnaire were the reasons for the successful outcome of this thesis. I would also like to thank my four interviewees Laura Pippard Vigander from Bærum kommune, Laura Perales from itslearning, Brian John Gelzhiser from Teleplan and Mili Orucevic from Visma for their time and participation for the interview and all the respondents of the questionnaire.

I would also like to thank my family, friends and my husband, Abdullah Moinuddin for their relentless support throughout my work.

# Table of Contents

## List of Figures

## List of Tables

# Chapter 1

## Introduction

Software testing is a broad aspect that keeps emerging in different stages of software development with different goals. Based on the ever-growing need and application of software products from day to day life, to mission critical systems, software testing is one of the most challenging and inevitable issues for companies, organizations, researchers etc. [1]Testing is used to ensure software quality in verification and validation and reliability estimation of software products. It is the most widely used approach to ensure software quality since software quality assurance occupies most concern in the software industry. When it comes to software testing, there are many factors to consider for software quality assurance. Whether the selected testing techniques are correct for the specific software, which testing types/methods need to be used, what are the software requirements, what technique needs to be used for validation and verification, how to do the code coverage testing, whether to keep documentation of testing processes etc., are all essential aspects needed to consider to approach software testing.

To facilitate the testing purposes and to meet the essential aspects of software testing, manual testing is no longer considered as the sole way of testing software products. Different types of testing tools for different aspects of software products have been introduced. It can be said that manual testing is slowly being replaced by testing tools for more test accuracy, less time consumption in testing etc. Testing tools differ from web testing tools to GUI testing tools, from Unit testing framework to cloud services. Not every company uses the same testing tools for the same purpose. It depends on the requirement of the software, the working structure of the company, developer's experience, budget of the project etc. There are some free testing tools available on the Internet and some testing tools need license to operate. There are companies who also build their own customized testing tools or customize standard testing tools for software products, projects etc. [2]Despite of the availability and advantages of using testing tools, not all software companies in Norway are embracing this change.

## 1.1 Background

Testing tools have made quality assurance of products rather easy for the workers. Different software products with a variety of requirements, need quality assurance testing in every level of software development. So testing tools make quality assurance of a software product less time consuming and more accurate than ever. Like any other software companies, Norwegian software companies are also using software testing tools for quality assurance of software products. But there is no specific information on how many companies are using these tools, and in which level.

A software application needs to be developed fulfilling all the requirements from the customer containing all the necessary functionality. But before releasing the product, testing needs to be done to make sure that the end product is bug free and contains all the necessities. The testing phase started as manual testing, writing all the test cases manually. The validation and verification processes were also manual; whether it was fulfilling all the requirements or developing a high quality end product. But for large scale projects, where test results and test cases need to be saved and repeated, manual testing has been seemed as a time consuming laborious task. With automated testing, testing has become easier and less error-prone. Tools used for validation and verification processes cover facts from meeting all the requirements of a product to delivering a high quality end product. Though it is still expensive to build such tools and prepare them for testing, for large scale projects, it is truly a lifesaver. Since then, many companies, organizations, business industries etc have embraced automated testing for software in their system. But in the real world, how many companies are actually using tools for testing? Are they using these tools after developing the product, during the development, or both? What kind of tools are they using in each step of development? Is the time-consuming process of setting up the tools for testing purpose, not to mention it's high costs of building, worth it? There are also some lingering questions like how the follow-up of a delivered end-product is carried out when it is associated with production and when it is associated with a project. Because when a software is developed in a manufacturing facility, then there are always place for software upgrades, updates and versions. But if a software is associated with project boundaries, then after delivering the end product, there is no way left for the developers to update or upgrade that product. Then this responsibility has to be carried out by some other departments. While talking about automated testing, it can also be considered as a possibility, whether there are ways to automate the writing of test cases instead of doing it manually. [2]These inquiries sums up to two main research questions of the specialized project which is the basis of this thesis report about this situation:

● What is the state-of-the-practice within automated software testing in the Norwegian software industry?

● What are the best practices of using automated tools for software testing?

These research questions was answered through a survey with the help of collected data from a questionnaire in the specialized project. The specialization project which is the basis of this thesis report was associated with automated testing in NAV, covering the automated testing facts, frequency of usage of such tools, cost of using such tools, outcome of using these tools etc. in NAV. But due to some circumstances, the specialization project scope had been widened not being confined only to NAV but also to other companies of Norway which are associated with software products. Investigation of the use of test automation in software quality assurance in Norwegian companies and organizations represents the usage of different kinds of testing tools in software quality assurance in different types of norwegian companies and organizations. For the investigation part, a questionnaire has been made following agile methodology to collect response from different types of norwegian companies and organizations. It was put on LinkedIn, sent out as email etc to get responses. The quantity of responses was not adequate with respect to the amount of send outs because the response time of the questionnaire was short. The responses were analysed using statistical analysis. Even with the few amount of responses, the answers of each questions of the questionnaire turned out to be quite diversified. And, the outcome of the specialization project reflected that most of the software companies of Norway are using manual testing as their software quality assurance factor and some other detailed information related to it.

## 1.2 Research Question

Testing tools can be upgraded, reused for the similar tests, efficient in testing purpose etc. Still they are not that popular among the software industry in Norway, the way they should be. These issues yield questions like:

*RQ1.* Why is the usage of testing tools still not that popular in the Norwegian software industry?

*RQ2.* How is it going on doing manual testing while following agile methodology for developing software/project?

## 1.3 Research Method

We have used survey methodology following quantitative research method to develop a questionnaire in agile way to investigate the use of testing tools in Norwegian companies. The data analysis method has been done using qualitative research method. To backup the questionnaire response rate, we have also developed a set of questions for interview and have conducted four of them to get the clear perspective of this investigation.

## 1.4 Contributions

We have made a questionnaire in agile way following quantitative research method. Then we have made four sets of samples of the questionnaire to send through email to the CEO and other Staff groups of IT companies, non-IT companies and universities. For this, we have made an Excel sheet documenting the name, type and contacts of 185 Norwegian organizations containing more than 20 employees. To backup the findings from the questionnaire sample, we have also made a interview question set relevant to the research questions.

## 1.5 Gist of the work

This project is the expansion of the specialization project work which was about the use of testing tools in software quality assurance in Norwegian companies. This time the sample of the survey has been modified and made more precise to find out the answers related to the research questions. Then it has been sent out to 185 Norwegian software companies which has more than 20 employees. The collected data obtained from the questionnaire are statistically analyzed to reveal the reasons about testing tools not being that popular and other facts related to this. To support the questionnaire, four interviews have been carried out to get real life information about the facts related to the research questions and the answers from them have been analyzed and linked with the questionnaire responses to get more understanding of the concept.

## 1.6 Thesis Structure

This project outlines as follows:

**Chapter 2** is about the theoretical terms related to this project to help understanding the thesis work.

**Chapter 3** reflects about the methods followed in this project to yield the outcome.

**Chapter 4** consists of the data collected by using the methods from earlier chapter and a short analysis of the specialization project's questionnaire

**Chapter 5** highlights the analysis, evaluation and discussion of the results.

**Chapter 6** consists of conclusion, reference and appendices.

# Chapter 2

## State of the art

### 2.1 Validation and Verification

Validation and verification are two important terms used in the field of software testing. The former one depicts whether the right system is being built, and the latter one inquires whether the system is being built right.

Validation is more likely to check whether the end product is meeting business needs and is fit for use. This type of testing is done by executing the software and checking whether the client requirements are met or not. It ensures the functionality of the end product, not the quality. Verification is to ensure high quality of the product, but it can not determine whether the end product will work or not. It ensures whether the software meets the specifications or not. This type of testing does not always need execution of the system and thus includes static testing techniques.

There are a number of techniques for software validation and verification (V&V). But there are no specific knowledge about using any particular technique for any particular event. [3]So there is an ongoing research about integrating V&V knowledge and knowledge management activities (like search, dissemination, use support) into a software engineering environment (SEE).

#### 2.1.1 Formal Methods

Formal methods are mathematically-based techniques or tools useful for developing either software or hardware. They are used for validation and verification of mission-critical system to ensure flawless performance. [4]In 2007, Constance L. Heitmeyer, Naval Research Laboratory from Washington DC, introduced a formal requirement method called SCR(Software Cost Reduction) which was used for the construction of requirement specifications and design documentation of the flight program of the U.S. Navy's A-7 aircraft. The SCR method did prove to be satisfactory for critical application properties by providing a practical formal method for  high quality requirement specifications and such. [5]In 1997, NASA published a case study of using formal methods as a tool for Independent verification and validation. They used the term 'lightweight' formal methods to do partial analysis on partial specifications while avoiding the lengthy way of establishing complete, consistent formal specifications.

### 2.1.2 Syntactic Methods

Syntactic methods are techniques that are used to detect and prevent bugs in applications from a syntactic level. This method is used for validation and verification of non-mission critical systems. It is a simpler and cheaper alternative of formal methods. This method works with the help of an abstract dependency graph which is created from the non-mission critical system. The developer can detect syntactic anomalies with the help of the graph in the system.

### 2.1.3 Test Cases

A test case is a set of conditions that testers apply to establish whether the functionality of an application is working as per requirements. It contains a set of test inputs, execution conditions, and expected results and post-conditions developed for a particular objective to verify compliance against a specific requirement. [6]In 2013, Saswat Anand et al. discussed automatic test case generation techniques including symbolic execution, model-based, combinatorial, adaptive random and search-based testing to investigate the discovery of new ways of test case generation. The paper offers a critical review on a number of existing automatic test case generation techniques by doing an orchestrated survey. An orchestrated survey means collaborative work collection of world-renowned active researchers on a key topic, in this case automatic test generation techniques. Their research concluded some facts like

- ART(Adaptive Random Testing) is considerably better than RT(Random Testing) with respect to F-measure(Expected number of test cases required to detect the first failure) and P-measure(Probability that at least one failure is detected by the set of test cases).
- Search-based testing is widely applicable.
- Combinatorial Interaction Testing (CIT) has a promising future research direction with refining sample space with the combination of program analysis technique and more.

### 2.1.4 Configuration Management System/Issue Tracker/Backlog

Configuration management system consists of all the building blocks such as source code, test scripts, third-party software, hardware, data and both development and test documentation of an application. It helps to ensure careful and thorough management of these building blocks during the entire life cycle of that application. It also allows to keep record of what is being tested. It is a complex system, so advanced planning is needed to document all the key features and their implementation in the development stage of the application.

### 2.1.5 Quality Manual

A quality manual consists of the quality system of the organization and its operational procedures. [7]A system needs to have a quality manual to achieve ISO 9001 certification so that documented information can be verified against the standard. ISO 9001 has four key requirements for establishing documentation:

- Quality policy: The approach the organization has for quality.
- Written scope: The key elements of the quality management system and also any part of the Standard that can not be applied with justification.
- Measurable quality objectives: What is the plan to achieve, how it will be measured or assessed if it is achieved or not etc.
- Procedure: Including all the operational procedures in the organization as flowchart, document or any media or format that works for the organization.

## 2.2 Software Quality Assurance

Software quality assurance means making sure of the quality of the delivered product that the product is defect, error and bug free. [8]ISO 9000 defines as "part of quality management focused on providing confidence that quality requirements will be fulfilled". It also means to ensure the quality in the software development process. It includes activities like process definition and implementation, auditing and training. Processes involves Software Development Methodology, Project Management, Configuration Management, Requirements Development/Management, Estimation, Software Design, Testing etc. After the implementation of these processes, software quality assurance has activities like identifying the defects in the software and taking necessary measures to resolve them. It not only ensures the fulfillment of all the requirements from the clients, but also makes sure that all the functionality of the end product are working properly
without any defect or error.

The purpose of testing software products is to assure the quality of delivered product to the customer. It is like an invisible commitment that has been made from the developers to the customer to deliver flawless, high quality products. But there are a number of different thoughts on this issue.

Some think that high quality products associate with high cost. Only projects with a costly budget can deliver a high quality product. Other speculations are that software products can never be flawless hence a specific level of defects can be accepted as normal.

There has been a number of works on software quality assurance for web applications. [9]In 2012, a model has been proposed for this purpose which has three parts: a server side, a client side and a server-client intersection side. This model has lessen the critical job of maintaining quality assurance of web applications in an efficient way.

### 2.2.1 Code Review

Code review is a well known practice that takes place during the software development phase to review code by the authors of code, peer reviewers and QA testers. This process helps finding errors, and fixing them at this stage is inexpensive and efficient for the organization. [10]In 2013, a research paper disclosed some facts about modern code review after surveying and interviewing developers and managers at Microsoft. One of the interviewee's comment had summarized many of the responses:

“[code review] also has several beneficial influences: (1) makes people less protective about their code, (2) gives another person insight into the code, so there is (3) better sharing of information across the team, (4) helps support coding conventions on the team, and [...] (5) helps improving the overall process and quality of code.”

### 2.2.2 Code Inspection

Code inspection is the most formal way of reviewing code. [11]It was introduced by Fagan in 1976 based on line-by-line group reviews in extended meetings. [12]Though it has been proven very helpful to detect defects in code by many researchers from time to time, the time-consuming lengthy structure of this approach has failed to be popular in practice. Code inspection is led by forming a committee of inspection leader, recorder, reader, author and inspectors. It is based on rules and checklists and makes use of entry and exit criteria. All the members of the inspection committee have a defined set of roles. During inspection, the documents are prepared and checked thoroughly against standards and checklists by the inspectors before the meeting, under the supervision of inspection leader. The inspection meeting generally takes two hours to finish while a reader presents the reports from inspectors. Then the defects are classified and recorded as a list and is sent to the author for rework. The follow-up meeting of rework is verified by the inspection leader.

## 2.3 Test Levels

There are a number of test levels. These test levels are important to perform, based on where they are put during the software development life cycle, or the level of test specification. Unit Testing, Integration Testing, Component Testing, System Testing and Acceptance Testing are the common test levels.

### 2.3.1 Unit Testing

Unit Testing is like testing the building blocks/units of a software product. Test cases in this level are normally written by the developers who writes code for the product to test a set of code or a specific function whether it can operate independently or not. Such as testing a function, a loop or a statement in a program is called unit testing. It is more viable to do unit testing using tools rather than manually, to save time and effort. Also using tools for unit testing will bring more correct results than doing it manually. E.g., if the submit button of user login form is not working, then a unit testing tool can quickly fix this with its pre-defined set of functions and syntax rather than fixing it manually. [13]In Microsoft, the developers switched to NUnit automated unit testing framework from individual unit testing. After a year of utilization of this practice on version 2 of a product, it showed greater results like lower defects in the end product and decrease in the development time.

### 2.3.2 Integration Testing

Integration Testing is testing the modules or integrated units of a software product to verify whether they are functioning correctly or not. Though all the parts of that module have already been unit tested, integration testing is still important to check whether the combined parts works well together or not. E.g., checking if an application testing the user registration form working fine with the user login form, is called integration testing. It can be done as top-down approach, which is starting with the top hierarchy and then going down within branches, or bottom-up approach which is starting from the lowest level of the hierarchy.

### 2.3.3 Component Testing

Checking the handling of data passing between different units and components is the main task of Component Testing. It is also known as module testing. The component testing always occur before integration testing to make sure the modules to be integrated are functioning well. Such as, if there are two forms named user registration and user login in an application, then component testing would be testing both forms individually to make sure they are functioning.

### 2.3.4 System Testing

System Testing or end-to-end testing is testing the almost finished product to check for its functionality as a whole, fulfillment of all the requirements etc. Such as while testing an application for system testing, all the key requirements and features will be tested instead of checking code, functions etc. Before reaching system testing, the application has to pass through unit, component and integration testing successfully. [14]For dynamic web testing, validation is important in the system testing. So an automated validation system has been proposed for WEB 2.0 application development.

### 2.3.5 Acceptance Testing

To test the operational readiness of the product as a part of a QA management system before final delivery, acceptance testing is done. This type of testing is mainly used in software development and software maintenance project. To check whether all the requirements are met, the system's user-friendliness, functionality etc., acceptance testing is necessary. Such as, in a user registration application, it will be checked for whether the buttons are in the right positions, text boxes are wide enough to type, how easy it is to navigate the application etc.

#### 2.3.5.1 Alpha Testing

Alpha testing is a type of acceptance testing that is carried out in lab environment by the internal employees as testers of the organization. It is done when the end phase of the software development stage nears. The aim of this testing is to carry out the tasks that a typical user would do. It is done before the release of the end product to identify all the issues and bugs in an early stage. It involves black box and white box testing techniques. All the bugs and issues that are found in the end product are dealt immediately by the developers.

#### 2.3.5.2 Beta Testing

Beta testing is also a type of acceptance testing but the testing is done in real time environment by the real users. It can be considered as user acceptance testing. It is done as the final testing before product shipment and a limited number of end users perform this test. It helps to obtain feedback about the product quality before release and ensures if the end product is ready for the real time users. It uses black box testing methods. Most of the issues that are addressed from the user's feedback are dealt in the new version of that end product.

## 2.4 Test Types

Some of these test types are typical for traditional software processes.

### 2.4.1 Regression Testing

When there is a modification in the code due to the changes of the requirements or functionality in the software, regression testing is done. It is to check whether new changes in code, previously fixed faulty functionality has come back or not. [15]A corporate research organization ABB conducted an interesting research of using automated regression testing in real life projects. Two real-life projects has been used in this regard which were associated with automated testing tools and it helped to list the common pitfalls, their mitigation plans and last but not the least how to avoid these pitfalls.

### 2.4.2 Compatibility Testing

This type of testing is done to check whether the software is compatible with other software application, operating systems etc. Cross-browser compatibility issue is a known problem with web applications as there is a new browser every now and then. [16]To address this problem with checking the web application behavior across different browsers, an automated solution has been proposed.

### 2.4.3 Acceptance Testing

This type of testing is done by customers and also known as User Acceptance Testing. Between any two phases of software development, this testing is done by end users on their own hardware.

### 2.4.4 Functional Testing

Functional testing tests specific types of actions or functions in a system. It also holds the answers of which features are going to work, what kind of functions the user can access etc. This testing is done according to the requirement documentation provided by the clients or design specification like use cases provided by the design team.

### 2.4.5 Non-functional Testing

Non-functional Testing goes for the quality of the software product. It tests the performance or behavior of the system under specific constraints.

**Table 2.1:** Functional Testing VS Non-functional Testing [17]

| Functional Testing | Non-functional Testing |
|---|---|
| 1. Functional testing is about what the system performs. | 1. Non-functional testing is about system's response or how the system performs. |
| 2. Functional testing is based on client's requirements. | 2. Non-functional testing is based on client's expectations. |
| 3. Functional testing includes Unit Testing, Integration Testing, Smoke Testing, Sanity Testing, Regression Testing and so on. | 3. Performance Testing, Stress/Volume Testing, Security Testing, Installation Testing etc. are categorize as Non-functional testing. |
| 4. Functional testing is about validating the behavior of an application. | 4. Non-functional testing is about validating the performance of an application. |

### 2.4.6 Usability Testing

Usability testing is to test the user-friendliness of the system. All sorts of usability functions like how the user interface is functioning with the application, how easy it is to navigate the application etc., are checked through this type of testing. Automating usability testing is quite tough compared to others as it is normally done by humans. Doing this type of testing at the end of the project leaves a risk of not having enough time to work on usability issues, and doing it early by human testers will only leads to inconclusive results. [18]So these kind of issues has been addressed in a usability testing framework named HUIA, experimenting it on Handheld device application(HDA).

### 2.4.7 Security Testing

To ensure the safety and confidentiality of data from the hackers, security testing is needed in almost every software product. With the increasing number of mobile apps and easy ways to download them on a mobile phone can cause security problems if the app market is not doing enough security check before letting them to be uploaded into the web. Security testing is applicable to all types of applications and systems. It is of a non-functional nature. This way malicious apps can take places in the market compromising with the security of app users. [19]So M. Riyadh et al. propose an Android-specific program analysis technique capable of generating a large number of test cases for fuzzing an app to test the security and robustness of Android apps.

## 2.5 Testing Methods

### 2.5.1 Static Testing and Dynamic Testing

**Table 2.2:** Static Testing VS Dynamic Testing[20]

| Static Testing | Dynamic Testing |
|---|---|
| 1. Static testing is about reviewing and examining the software without executing it. | 1. Dynamic testing is testing and using the software while executing it. |
| 2. Static code review, inspection etc. techniques are used in static testing. | 2. Unit testing, integration testing, system testing and non-functional testing like performance, security testing etc. techniques fall under the category of dynamic testing. |
| 3. Static testing is about prevention of defects. | 3. Finding defects and fixinf them are the main principles of dynamic testing. |
| 4. Static testing does verification process. | 4. Dynamic testing does validation process. |
| 5. This testing involves checklist and process to be followed. | 5. This testing involves test cases for execution. |
| 6. Code reviews can be started in an early stage of project so static testing is cost effective. | 6. Dynamic testing can not be started until a certain stage of the project so it is not that cost effective. |

### 2.5.2 White Box Testing

White box testing is about testing the internal structure of a software which is inaccessible to the end user. It requires an adequate amount of programming knowledge to do this type of testing. Basically the developers or a separate testing department of a company carry out this testing. The structure/code of the program is as important part for this type of testing. [21]Random mutations on pre-defined inputs of a program and comparing the resulting output is a useful way to find security vulnerabilities in software and fuzz testing tools has shown its success by using this technique. A code-coverage maximizing heuristic algorithm has been incorporated with a tool named SAGE(Scalable, Automated, Guided Execution) to enable white box fuzz testing to find defects of Windows applications without any hassle.

### 2.5.3 Black Box Testing

Black box testing is more about functionality of the program. It does not require to have knowledge about the internal structure or code of the software. Independent testers can do this type of testing and they do not have to be technical experts. This type of testing focuses on what is performing or carried out. [22]AutoBlackTest is a technique that focuses on testing interactive applications. It automatically generates test cases in the system level of the application. The main technicality is its Q-Learning, which enables AutoBlackTest to learn interaction with the application under test and stimulate its functionality.

### 2.5.4 Grey Box Testing

Grey box testing is a combination of white box and black box testing. It comes with limited information about the internal structure of a software. The testers needs some required knowledge about coding as this type of testing will be provided with detailed design documents along with requirement related information of clients.

## 2.6 Automated Testing

Automated Testing is a testing process where software testing tools conducts pre-scripted tests on software to verify whether all the functionality are working properly, all the requirements for the software application are met properly, the version of the software application is bug free and updated etc.

### 2.6.1 Why Automated Testing?

With the advent of testing tools, automated testing has become more and more popular. There are a lot of online software testing tools available in the Internet. Companies also use customized testing tools for their software which takes time, effort and money to build. But for QA control, to be able to run tests repeatedly, to use same type of tests and test results later, and to compare test results automated testing tools are inevitable. One can use one testing tool for all test types and levels, or different ones for different types and levels. Regardless of the size of company or project, the use of automated testing is increasing because it makes testing easy and helps to deliver almost flawless end product in shorter time period. [23]A research from 2009 depicts that, large scale company like Microsoft is using NUnit automated unit testing framework for unit testing purpose. [15]On the same context, not so large scale company ABB, has worked on automated regression testing since 2004.

#### 2.6.1.1 Time and cost

Automated testing is not only good for test efficiency and QA, but also for test speed. It takes less time to do the testing of a software application using a testing tool. Since time is money, the less time it will be needed to do the testing in a project, the faster will be the product delivery making it cost effective. Also, automated testing tools can help to reduce time and cost to run repetitive tests.

#### 2.6.1.2 Helping hand for the workers

Testing tool have been a great helping hand for the developers, testers and other workers associated with the development of software applications. Developers can know about errors and fix them even before QA session by running automated testing whenever there is a modification in the source code. Testing tool helps testers to do their testing more effectively without worrying about errors that may lead to major problems. In a word, not only for bug-free end product but also for safety measures, testing tools helps the workers reduce unnecessary stress and workload providing greater test coverage.

## 2.7 Manual Testing

Manual testing has been performed on the first software product that needed to be tested. To find any defects, bugs in the product, or functional problem due to some requirements, test cases are written to cover these kinds of inconveniences before the delivery of final product. The tester goes through the functionality of the product as an end user to find any faults/errors and perform testing following a certain test plan.

Whenever there is any change in the product requirements, that change can be easily tested in the product by writing new test cases for that change. But it also takes time to write test cases and do the testing manually when it comes to large-scale projects. Also when it comes to repeat tests or use previous test results, manual testing can be a time-consuming, laborious method.

### 2.7.1 Automated Testing VS Manual Testing

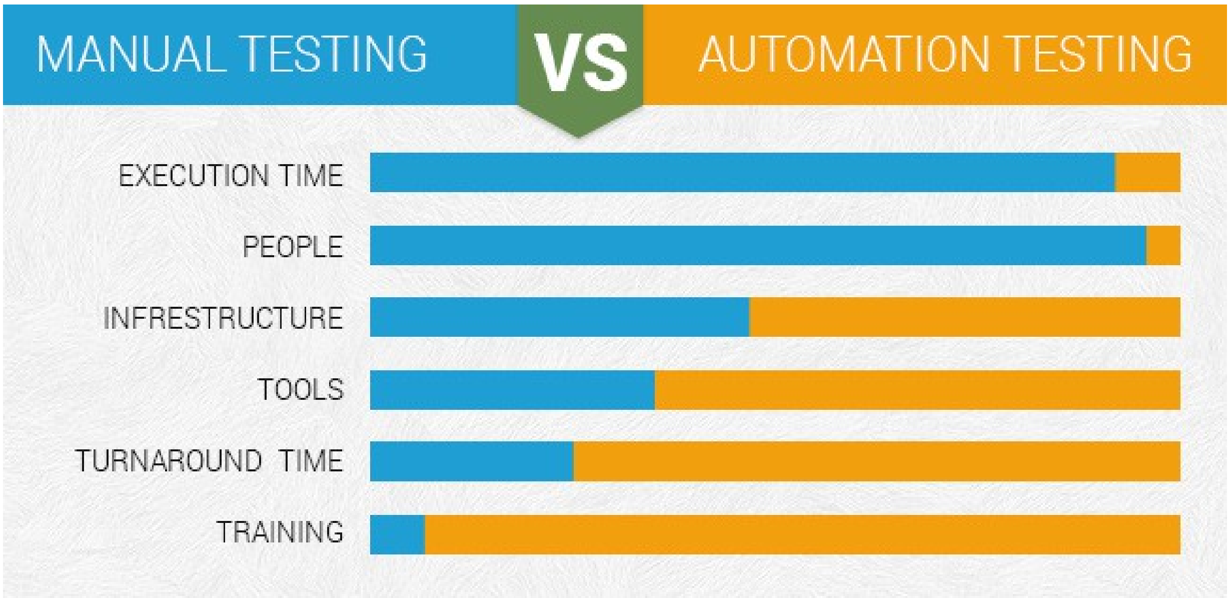


**Figure 2.1:** Automated Testing VS Manual Testing[24]

There are a few factors like test execution time, people needed to do the testing, infrastructure of the test, tools, turnaround time of the program and training- which differ from manual testing to automated testing. From Figure 2.1[24], it is easily understood that most people are needed to do the testing manually, whereas most training is needed to do automated testing.

## 2.8 Test Tools

There are a number of different kinds of tools for testing used in various situations.

### 2.8.1 Automated Web Testing Tools

This type of tools are used for testing web applications to discover bugs from new versions of the application. Selenium, Watir, Windmill, Ranorex etc are the most popular automated web-testing tools for web applications. Selenium is a well known web testing tool, but it has its limitation of not being able to be used in all web browsers. [25]U. Nidhika and C. Vinay have proposed a solution of this problem in their paper by suggesting integration of Selenium IDE with a web driver in a single package.

### 2.8.2 Automated GUI Testing Tools

For desktop applications it is hard to do robust GUI testing. A change in the user interface can break apart the structure of the whole application. So, automated GUI testing tools like Squish, Ranorex, TestComplete etc are there to help for doing robust GUI testing.

There has also been some work for GUI testing in dynamic web pages. [26]TestComplete, which is widely used for software testing and showing excellent performance for Windows,web or client software, has been modified to meet the need for GUI testing in dynamic web pages.

### 2.8.3 Unit Testing Framework

Unit testing frameworks and tools are widely used for unit testing to make it easy for the developers. NUnit, xUnit.net, JUnit etc are some popular frameworks used in unit testing purpose. Test case prioritization is to prioritize the important tests to execute first. [27]D. Hyunsook et al. have come up with a controlled experiment in their paper of effectiveness of test case prioritization on Java programs tested under JUnit. Through their experiment, it has been shown that test case prioritization has been proved beneficial for JUnit causing more fault detection.

### 2.8.4 Automated Testing Cloud Service

This type of service allows to do testing in different environments and on different machines without building any specific testing infrastructure. [28]Sauce Lab, TestingBot, Gridlastic etc are popular for this kind of service.

## 2.9 Code Coverage and Test Coverage

Code coverage is about determining how much code of an application has been tested. For example, there is a code path that can only be executed during an error condition. So, the developer needs to write a test case that would create that error condition and then verify that the appropriate error message has been displayed. Code coverage utilize white box testing technique to assess the amount of coverage by going through source code. Statement coverage and block coverage, function coverage, branch coverage and modified condition/decision coverage are the main types of code coverage analysis. A number of testing tools for code coverage purpose are available online. [29]In 2014, Khalid Alemerien and Kenneth Magel did a controlled experiment to find out how consistent is the measured value of coverage metrics of statement, function, branch and line coverage with different code coverage tools. Their paper summarizes that branch and function coverage metrics are significantly different. Statement and line coverage has the most similar results. So the results of some code coverage tools can be misleading.

Test coverage defines the measurement of the amount of testing performed by a set of test. It ensures that the test suites are actually testing the application code. It finds out the requirements that are not covered by a set of test cases, helps to create additional test cases to increase coverage and also identifies the meaningless test cases to eliminate. It can assure the test quality of an application and can prevent defect leakage at an early stage of product life cycle. [30]In 2009, Audris Mockus et al. did a multiple case study on two dissimilar industrial software projects to find out about the test effectiveness of test coverage. Their work also reflects the relationship between test effort and the level of test coverage. The results show that the increase in test coverage is correlated to the decrease in field reported problems when it is modified for the number of pre-release changes. The drawback of increasing test coverage is that it occupies more time so achieving complete coverage is impractical and inefficient.

## 2.10 Requirement Traceability

[31]Traceability is an important element of documentation of code of an application that indicates the extent to which it can be traced to its origin. An application is said to be traceable if it complies with its specification. [32]Requirement traceability is “the ability to describe and follow the life of a requirement, in both a forward and backward direction”. It is useful for finding out which specifications of an application have made it to the requirements and later to the design artifacts, code and test cases. [33]A research paper highlighted the main problem of requirement traceability and that is it does not have single modeling method. Hidden information, complexity in information, immature integration technology, lack of commitment from all parties etc can also create problems in requirement traceability. The author suggested using realistic techniques like information retrieval or using traceability matrix etc. as solutions for these problems.

### 2.10.1 Bi-directional Traceability Matrix

A bi-directional traceability matrix is the combination of forward and backward traceability. It means this matrix maps requirement to test cases and vice versa. It checks whether the project is progressing in the right direction and for the right product. Each requirement is tested thoroughly whether it is applied to the product or not. It also verifies whether the product is in the right track and there is no unnecessary elaboration in the scope of the project by mapping the test cases to the requirements.

### 2.10.2 Tools(TBReq, SmartExcel4TFS)

There are a number of tools available online for requirement traceability. Some of them are licensed and some are free. [34]TBReq is a licensed requirement traceability matrix tool that is a part of the LDRA testing suite. It is highly configurable and user friendly. It links requirements, design, development, testing and verification work-flow to the requirement management tool very easily. It also generates graphical traceability reports which is also easily exportable to Word documents. SmartExcel4TFS is also a licensed requirement traceability matrix tool from Microsoft TFS. It tests the user requirements to be bug free and keeps count on requirement coverage. Work items can be updated from Excel and color coded concise report can be formed with the help of this tool.

## 2.11 Software Development Process

A software development process is the method through which software is developed based on the project's aim and type. There are a number of development processes or models that are developed over time to meet different needs of different projects. The selection of the development process is very important as the wrong selection of the model can hinder the success of a project. There are agile, waterfall, spiral/incremental, test driven and DevOps or combinations of these development models available to be utilized. Among these, agile method is still going strong with its universal popularity. DevOps is comparatively new and also giving the agile methodology a good competition. The choice of development model also impacts the testing techniques.

### 2.11.1 Agile Methodology

The agile methodology is a type of incremental development model but promotes a lighter and more people-centric viewpoint than traditional approaches. Software is developed in iterative, rapid cycle e.g. using the Scrum framework, Kanban or other approaches. Each cycle, which is called sprint, consists of a small release of a portion of the software functionality and the activities of next cycle depend on the activities of the previous one. Each release is thoroughly tested to ensure software quality. It is a good method to obtain customer satisfaction by communicating them weekly with updates of software. The functionality of this method can be summarized by the following figure:

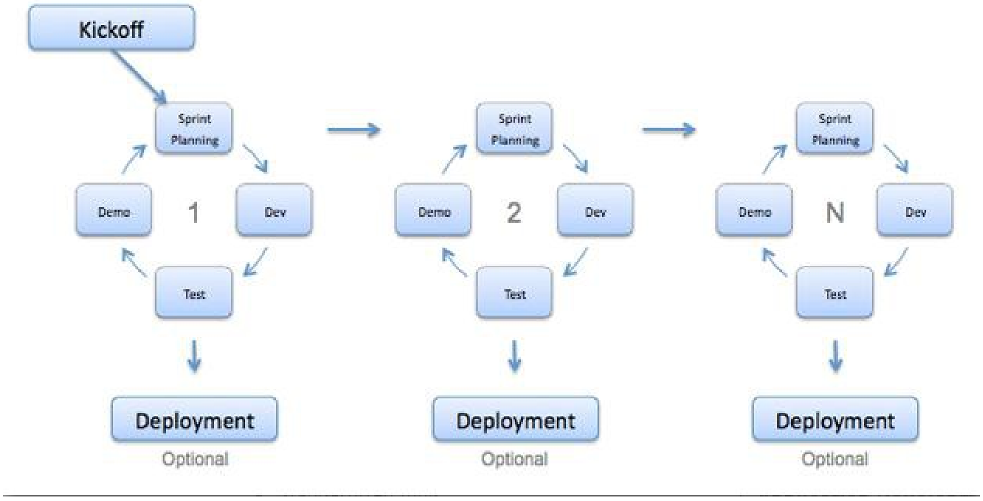


**Figure 2.2:** Agile Methodology[35]

[36]In 2007, an empirical study conducted in Microsoft revealed the popular use of agile methodology in practice. The study showed that one third of the respondents use this method for reasons varying from the flexible design of this method to improved communication among team members. It also revealed the practical scenario of using agile method for large scale projects where more than twenty team members are involved.

### 2.11.2 Waterfall Development Model

[37]The waterfall method is the first development process to be introduced by Winston W. Royce in 1970. It is also referred as linear-sequential life-cycle model. It is a simple linear method easy to understand and apply. Each stage of this process is independent and after accomplishing one stage, the next stage can be pursued. So there is no overlapping of stages. The stages in waterfall model is summarized in the following figure:

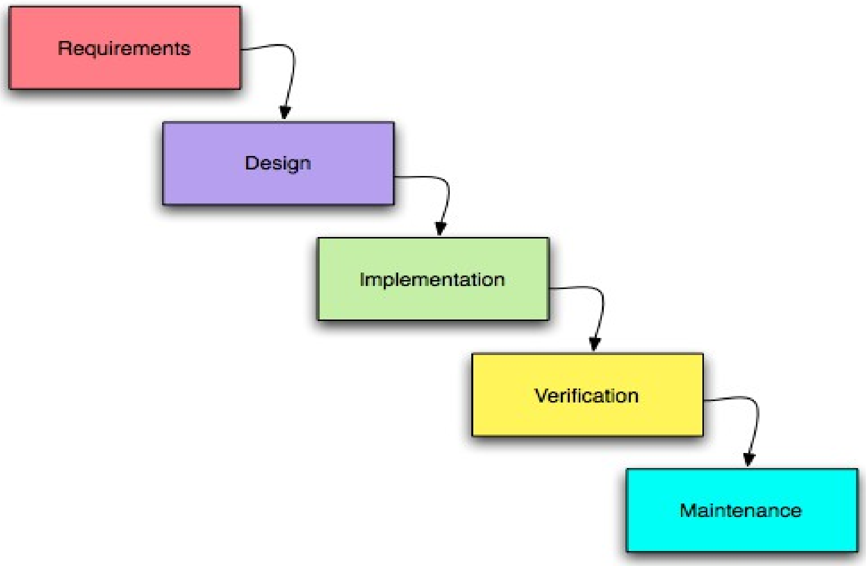


**Figure 2.3:** Waterfall Development Model[38]

[38]The requirements stage is for gathering all the requirements of the system to be built and making a requirement specification documentation. As per the requirements, the system is designed defining the system architecture in the design stage. The design stage is divided into two sub-phases; Logical Design and Physical Design. The logical design phase consists of the design of the system independent of any software or hardware specification, based on the information retrieved from requirements stage. Once the logical design phase is done, it is transformed into physical design specifying software and hardware technologies. With the help of system design, the implementation stage helps to develop the system following project requirements and specifications. The verification stage is there to ensure the fulfillment of customer requirements in the developed system. System modifications and bug fixing are done in maintenance stage where customers get to use the end-product and give feedback about their experience with the product. The waterfall model is less suitable for projects with long and ongoing life-cycle, high risk of uncertainty and not fixed requirements. [39]In 2005, Ericsson AB, a leading and global telecommunication company, moved to the incremental and agile methodology for their large-scale projects after facing critical problems with issues related to requirements and verification while using the waterfall model.

### 2.11.3 DevOps

DevOps is one of the most popular development process now- a-days. [40,41]It was first introduced in 2008 in an Agile conference which later became popular with the term DevOps in 2009 in Belgium. DevOps provides a number of processes and methods to collaborate among development, QA and IT operations and strengthen communication among collaborators. There are a set of Architecturally Significant Requirements or ASRs such as deployability, modifiability, testability and monitorability which are very important to be met by software in order to practice DevOps successfully. To meet these requirements, DevOps provides a set of tools or as it is called tool chain. Automated tools like Code, Build, Package etc. to develop an increasingly programmable and dynamic infrastructure. Companies that practice DevOps are accomplishing more work in a simple fashion by forming a cross-functional team of members from QA, developers, testing, operational analyst and so on. They not only have technical benefits like good communication, faster problem solving and continuous software delivery, they also benefit in the business by having stable operating environment and more time to do more.

### 2.11.4 Spiral Development/Incremental development

The spiral development model is a mix of iterative development process and waterfall model with the high significance on risk analysis. It has four stages: planning, risk analysis, engineering and evolution. A project passes through each stage in an iterative fashion or in this case, it is called spiral. The following figure summarizes these four stages:

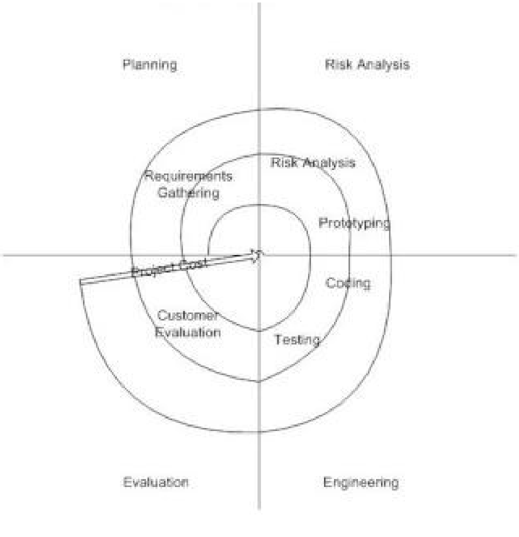


**Figure 2.4:** Spiral Model[42]

From Figure 2.4[42], it can be seen that the planning stage is where all the requirements are gathered including Business Requirement Specification and System Requirement Specification.

Risk analysis stage covers identifying risks and preparing alternate solutions. A prototype is developed at the end of this stage. If any risk is found then alternate solutions are applied. The engineering stage is all about developing the software and testing all the functionality. The output of the project is evaluated by the customers in the evaluation stage before the project goes to next spiral. When requirements are complex and the product line is new, cost and risk analysis is important, then the spiral model can be a valid option. But it is also a highly expensive model to use with all the specific expertise for risk analysis, so it is unwise to use such a model for small projects.

Iterative development model is about working in iterations or cycles. The start of a project using iterative development does not start with a full set of requirements. It is almost like identifying the requirements while developing the software. A part of the software is implemented which then will be reviewed to identify further requirements. This process continues until a new version of the software is developed in every cycle of the model.

### 2.11.5 Test Driven Development and XP

Test driven development or TDD is a test-first approach where test cases are written before the development of the software is done to fulfill that test. It is a way of planning and thinking through the design and requirements before starting to write the real functional code. The simple procedure of test driven development is summarized in the following figure:

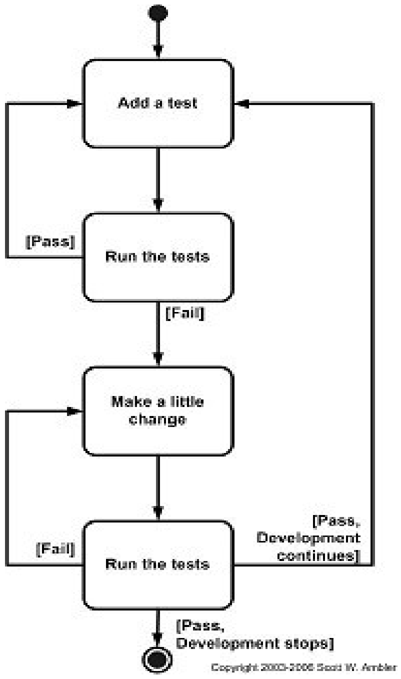

Add a test
[Pass]
Run the tests
[Fail]
Make a little change
[Fail]
Run the tests
[Pass, Development continues]
[Pass, Development stops]
Copyright 2003-2006 Scott W. Ambler

**Figure 2.5:** Test driven development[43]

[44]In 2012, a number of research projects and experiments have been conducted at universities and companies like IBM and Microsoft to find out about the effectiveness of test driven development. Some of the experiments in IBM and Microsoft revealed that TDD delivers more reliable code and better code coverage, but it takes more developers to work on projects following TDD method.
[45]XP or Extreme Programming is one of the popular agile processes that has been introduced in 1999. [46]It was proven quite successful to many companies due to its simple and clean design. Instead of long waiting time of product delivery, it has small but early delivery to customers and based on their feedback, developers work on the modifications. TDD or Test-driven development is a part of XP.

## 2.12 Test Documentation

Test documentation are developed before or during the development of the software. It envelops all the important aspects while developing the software like estimating the testing effort, test coverage etc. in a form of document to keep records for future. Keeping test documentation brings systematic approach to the testing process. Testers need the software requirement specification and functional documentation for regular implementation and modification. Test documents can be different types. Some of the mostly used documentations are enlisted below:

- Test plan: The test plan includes topics like which features of a project to be tested or not, what is the test approach, what are the deliverable(test cases, test reports etc.), list of phases and tasks with their scheduling etc.
- Test design and test case specification: The test design documentation contains a set of test procedures to ensure all the requirements of the software are met. The test case specification is about test scenario or conditions. A good test case is capable of finding more errors to keep the software bug free. Test case is designed based on test scenario and executed manually or automatically.
- Test Strategy: Test strategy describes the testing approaches the testers need to take to complete the software testing life cycle. It also contains roles and responsibilities of the team members. It contains the testing levels(Unit testing, Integration Testing etc.) and who will be doing what levels etc.
- Test summary report: After accomplishing all the tests, a test summary report is made. It explains all the activities and details of the testing that are done in a project to stakeholders like senior management, clients etc.
- Weekly Status Report: The weekly status reports are important to document weekly updates, issues and milestones for further weeks. It even helps to track the weekly performances of team members and prioritize future actionable items.
- User Documents/ manuals: It is a technical guideline or technical communication document which contains information and certain rules of how to operate a system for the users.
- Risk Assessment: The risk assessment document is to analyze risk and document them with proper solutions. Not all the projects impose the same risks. It is important to start analyzing risks during planning phase of the project. Not all the analysis would be correct, so it is advisable to re-assess these risks during the continuation of the project and adjust the solutions accordingly.
- Test Log: It contains information about whether a particular test case execution has been successful or not.
- Test data: It contains sample data for all sorts of testing.
- Bug Report: A written report that enlists the bugs or errors while testing a software is called bug report. An effective bug report highers the chances to fix the bugs in the software. The key points of writing a bug report is to have clearly specified bug number, describing the steps of reproducing the bug and summarizing the problems that the bug causes.

## 2.13 Related Works

Usage of software products is not confined to the computer anymore. It has become a common phenomenon in day to day life. It has evoked different requirements from the customers and software companies are trying their best to keep up with new requirements. Developing the software is like winning half the battle. There is no guarantee that a developed product is going to function properly meeting all the requirements from the clients, and ready to go live. Software testing is necessary to ensure the functionality, security and requirements fulfillment. [47]It is though interesting to know that until fifties, the primary testing method was debugging as there was no distinct difference between debugging and testing. Software testing was first mentioned in 1961. Later in 1969, the first static and dynamic testing tools were created using data flow analysis for improved test coverage. The first big move was in 1979, when Glenford J. Myers distinguished debugging from testing clarifying about various testing principles. It was during this time when software testing became more evident and it is going strong till this date. Software testing these days differs from manual to automated. Manual testing has always been there whenever the software products need to be tested for any defects, bugs or error. But solely depending on manual testing for large scale projects can lead to a time-consuming, resource-hungry testing process and not to mention error-prone.

To increase the efficiency in the testing process, to simplify the testing process and to get an error and defect free end product in less time, testing tools play an important part. [47]In 1985, AutoTester the first commercial testing tool was invented for the personal computer. Since then there has been a remarkable upgrade and variations in testing tools. Nowadays, automated testing has become an invincible part for the companies that develops software products. It has now somehow replaced the human testers for testing large parts of the software products. There has always been a question associated with testing regarding quality assurance of software products. The main purpose of various testing methods is not to compromise with the quality of the delivered product. [47]In 1979, the QA standard BS 5750 was published. It was later merged into ISO 9000. ISO 9000 is about quality management and quality assurance standards. Now ISO 9001, which is a part of ISO 9000 standards, is being widely used for its generalized and abstract guidelines.

There has been a number of works based on Automated Testing. From testing SQL vulnerabilities in web applications to GUI testing in web pages, automated testing has been tried and successfully implemented on various applications.

**Table 2.3:** Automated testing related works

| Topic | Description |
|---|---|
| 1. Avoiding SQL injections in web applications | [48]In 2007, a new technique was introduced to avoid SQL injections in Web applications by using syntactical and semantic knowledge of the SQL queries as testing purpose. It's called SANIA or Syntactic and Semantic Analysis for Automated Testing. Some real world web applications have been tried with this vulnerability technique and has been proven successful. |
| 2. Finding security vulnerabilities in software | [21]To find security vulnerabilities in software, fuzzy testing tools have shown its success by using random mutations on predefined inputs of a program and comparing the resulting output. A tool named SAGE(Scalable, Automated, Guided Execution) has been incorporated with a code-coverage maximizing heuristic algorithm to enable white-box fuzz testing to find defects of Windows applications in a jiffy. |
| 3. Easily availbale automated testing resources and services | [49]A cloud- based service named Automated Testing as a Service(TaaS) has been introduced to make resources and services for automated testing easily available for everyone. Continuous testing techniques benefit the developers by reducing the effort and time while producing high quality code from the start. It is mainly done by using the resources of the cloud, combined with the development environment. There also a number of selected services for the end users as well. |
| 4. Automated GUI testing | [50]Graphical User Interface or GUI is a much needed feature for any software application and successfully testing GUI with tools is a tough |

| | |
|---|---|
| | challenge. Though GUI testing are normally done manually as it is navigated by human, an automated GUI robustness testing tool has been proposed for applications running under MacOSX, which is a JAVA library called GUITest. There has also been some work for GUI testing of dynamic web pages. |
| 5. GUI testing in dynamic web pages | [26]A well-known testing tool, TestComplete, which is being widely used for software testing purpose and showing excellent performance for Windows, web or client software, has been modified to meet the need for GUI testing in dynamic web pages. |
| 6. Real-life practice of automated regression testing | [15]There has been interesting research done by the corporate research organization, ABB, about the real-life scenario of using automated regression testing in projects. It has taken two real-life projects using automated testing tools to list the common pitfalls, their mitigation plans, and last but not the least, how to avoid these pitfalls. |
| 7. Literature review and real-life practice of Automated Software Testing(AST) | [51]There has also been some work depicting the difference between literature reviews and practical experiences of AST(Automated Software Testing). Refactoring means rearranging in a way that preserves the behavior of a program like IDE while improving its design. |
| 8. Automated testing of refactoring engines | [23]To keep the refactoring error free, automated testing of refactoring engines has been proposed where the test input is generated and resulted output is checked, while automating test of refactoring engines. |
| 9. Testing framework for Handheld device | [18]Some interesting works on Handheld device |

| | |
|---|---|
| application(HDA) | application(HDA) has also been done using a usability testing framework named HUIA. |

# Chapter 3

## Research Methodology

A research methodology describes the ways to acquire necessary data and analyze them to produce the ultimate result. In this project, the necessary data was collected through survey methodology which is why a questionnaire has been made following the agile method. Then the collected data from the questionnaire has been analyzed using quantitative data analysis method and results have been obtained.

### 3.1 Survey Methodology

The survey methodology is one of the popular research methodologies. A survey can be anything from a simple handwritten form to an extensive interview. It consists of a sample like a development department of a software firm or bigger, method of data collection like questionnaire and individual questions to analyze collected data.

### 3.2 Quantitative Data Analysis

Quantitative data analysis is about the numbers, facts, statistics etc that can be retrieved from a situation like interview, questionnaires etc. Online questionnaires, on-street or telephone interviews are data samples for quantitative data analysis method. Highly structured and rigid techniques to get responses on pre-formulated questions are ways to collect data in quantitative analysis. Quantitative data analysis has conclusive, reliable population-based data which can be generalized in representative samples.

### 3.3 Qualitative Data Analysis

Qualitative data analysis can be described as a scientific research technique. [52]"In general terms, scientific research consists of an investigation that seeks answers to a question, systematically uses a predefined set of procedures to answer the question, collects evidence, produces findings that were not determined in advance and produces findings that are applicable beyond the immediate boundaries of the study." Qualitative data analysis involves all these features. One of the main characteristics of qualitative research is to describe the experience of people in a given research issue.

**Table 3.1:** Qualitative Research Method VS Quantitative Research Method

| Qualitative Research Method | Quantitative Research Method |
|---|---|
| 1. This research method is text-based. | 1. This research method is number-based. |
| 2. This method does not need statistical tests for analysis. | 2. This method needs statistical tests for analysis. |
| 3. It describes a problem or situation from the perspective of people experiencing the problem or situation. It is more subjective. | 3. It describes observed effects of a problem interpreted by researchers. It is more objective. |
| 4. It involves focus groups, in-depth interviews etc. | 4. It involves online questionnaires, on-street or telephone interviews etc. |

## 3.4 Agile Methodology

The agile methodology is an iterative and incremental process where throughout the development life-cycle one can go back and forth. The agile method has been followed in this project while developing the research questions and questionnaire. The questionnaire that is used in this project is an extensive modification of the questionnaire used in the specialization project. The modification process that has been carried on using agile method such as modifying questions with the options and submit the questionnaire to the supervisor every week for feedback. These weekly feedback and modifications in the questionnaire has helped to form the final version to send out to the software companies in Norway. The same process goes with the interview questions for this project. The interview questions are mainly based on the research questions and the responses of the respondents from the questionnaire.

## 3.5 Questionnaire

A questionnaire is a set of questions that can be used to do surveys or interviews. The questions can be produced based on different studies and reports. It should contain different kinds of questions to consider the wider aspect of any topic and to eliminate the monotonous pattern. There also should be some demographic questions where users can convey some information about themselves. It should have a time limit and not be too lengthy, so that the user would not feel bored and can pay attention while answering. It can be sent out as printed material or can be put into Internet for viewers response. It is always better to get more response from the viewers so that analysis of that questionnaire can be done thoroughly with more data. Questions in a questionnaire can be open ended or close ended or the combination of both. It is however evident that most of the time, close -ended questions are more answered by the respondents.

## 3.6 Project Method

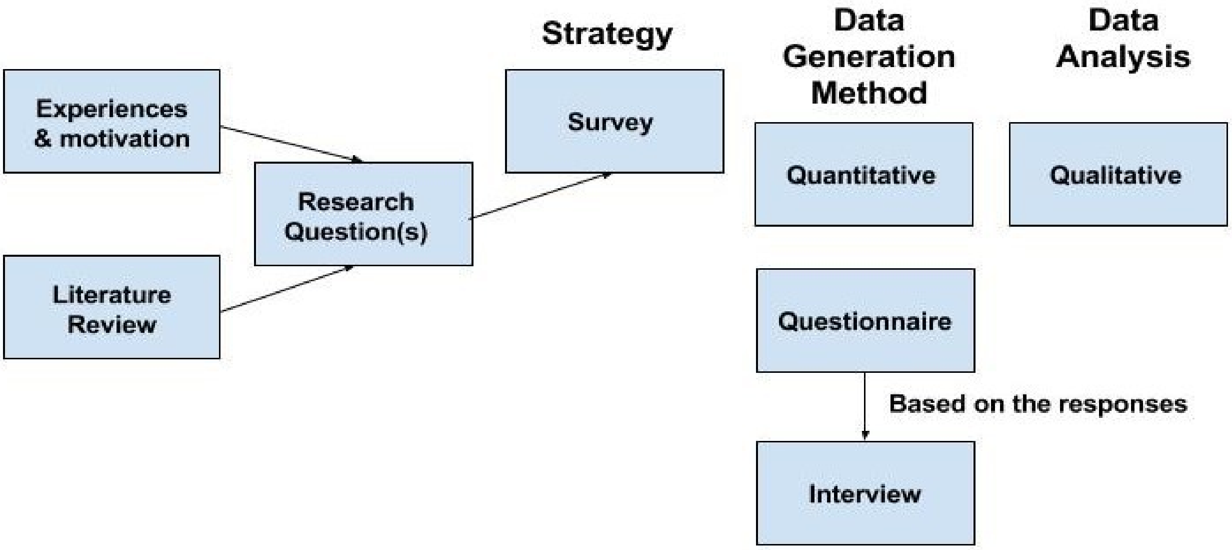


**Figure 3.1:** Research Methodology

The research methodology involves steps like experiences and motivation along with a literature review to come up with research questions. A course on software requirement and testing, and papers and articles based on automated testing helped a lot to form the research questions for the specialization project. Based on the outcome of the specialization project, we decided to do the thesis. For this line of project, the survey methodology has seemed proper. The strategy is to collect data on automated testing through a questionnaire and analyze them. Survey consists of a sample like a development department of a software firm or bigger, method of data collection like questionnaire and individual questions to analyze collected data. Automated testing is a broad aspect. Usage of testing tools in different companies of Norway is a somehow composed idea I have finally decided to work with. A questionnaire have been made based on how many organizations/companies are using testing tools, how it is going, what kind of industries are using these tools, how much budget they are holding for this etc. After sending out the questionnaire to the selected organizations/companies in Norway and based on the feedback from the respondents, a thorough analysis of the collected data has been executed. The data analysis method is qualitative. The formation of research questions and questionnaire have been done using the agile method following quantitative research method. Which is why, after a number of moderations and updates, the final research questions and questionnaire have been prepared. There has also been weekly meetings so that the research questions and questionnaire can be modified properly. Based on the responses from questionnaire, we decided to form a set of interview questions which also includes facts covering the research questions. Figure 3.1 holds all the key points from the above description.

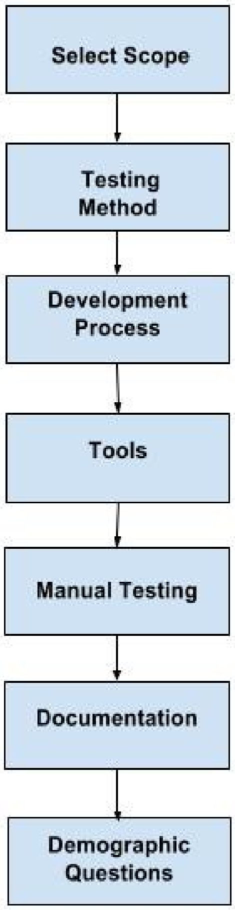


**Figure 3.2:** Structure of Questionnaire(Survey method)

The questionnaire is important to do a survey. From Figure 3.2, we can see that there is a number of sections like Development Process, Tools, Testing Methods etc.. There also is a decision making question about selecting the scope of the answers at first whether the respondent wants to answer based on project or product and so on. Each section holds a number of questions related to that section. A good number of open-ended and some close-ended questions have been used in this questionnaire. The demographic questions section also holds question about interested respondents who would like to do an interview. The testing method section is about which testing methods are used in the development of software products. The development process section contains questions about which testing processes are used in the development of software products. The tools section is about the tools that are being used in the testing of the software product. The manual testing section contains questions about manual testing is where one needs to write test suites for the software application to perform the test on them manually instead of using testing tools. Documentation section refers to the documentation of testing processes and tools.

This questionnaire has been sent out to a number of different companies in Norway, and also has been put on Internet for response. The more the number of responses, the better the data analysis would be.

The questionnaire sample has been made using Google Forms. We have made five separate link instances for companies who are solely software or IT based, companies who are partly using software or IT systems for their operations and universities. These links were separated by CEO group of software/IT companies, other employees of software/IT companies, CEO group of non-software/IT companies, other employees of non-software/IT companies and universities. These differences has been made to notice the difference in response based on the type of company and job role. And also not all the website or information directory database we have accessed, have email addresses of the CEO of the companies. We have also narrowed down the sample sending criteria to companies who have more than 20 employees such as middle and large-scale companies. We have sent the questionnaire samples through email to 185 organizations and also to some websites which do not contain any email address but have built-in contact forms. As per the questionnaire sample has five link types, the email also has been sent in five different groups. There is a question in the sample concerning respondent's interest for interview. We have prepared ourselves to do interviews as from earlier experience in the specialization project, getting a satisfactory amount of responses from questionnaire is quite uncertain.

# Chapter 4

## Data Collection

### 4.1 Data Collection from Interviews

Four interviews have been held from the responses of the questionnaire. Each of the interview contains one employee from different company with different job role. So we have got four interviewees from CEO and other staffs of IT/software and non-IT/software companies. The media of the interviews was Skype. All the interviewees have been asked the same set of questions and their answers have been quite interesting. We are going to address the interviewees as In.1, In.2, In.3 and In.4. The interviews with the questions and answers are summarized below:

*Q.1 When do you start QA in your projects? If you can elaborate on the process.*
*Ans.* In.1 has been following the Kanban process which is an agile process framework. So the broken down small tasks of a project are assigned to the Kanban board. Then the development of those small tasks are carried out. After that they will perform the QA and testing. For each small task there will be analysis, development, QA and testing before labeling it finished.
In.2 has been following PRINCE2 which is a process based approach to project management. So they start working on QA and test plan as soon as the project starts because of the mandatory business plan document.
In.3 is doing agile method for their projects and feels that QA should start quite early in the development phase. They prefer to continue QA in a parallel way with the development phase of the project but in practice, QA starts much later, probably after the end of the development phase in their company.
In.4 also have their development, testing and QA part continue simultaneously for their projects.

*Q.2 How popular is the use of testing tools in your company?*

*a. (if popular) Why do you think it is the way it is? What are the factors behind it?*

*I. Which tools do you use usually for your projects/products?*

*Ans.* In.1 revealed the popularity of testing tools in their company. They have older large administrative system which they are currently integrating and contributing to digitalization as many of them are still manual form system. They use JUnit for Java codes. For .Net projects they prefer NUnit and also some built-in tools that comes with Visual Studio. The reason of the popularity of testing tools in their company is that they get better products as they are well tested with the help of these tools. The testing takes less time than manual and there is far less chance of jeopardizing the source codes while using tools for testing. They are going to bring in Jenkins and SonarQube for their new projects and they are also going to work for more efficient code coverage with the help of testing tools.

In.2 is using some testing tools for every project. They are primarily using Team Foundation server in the building process and other build servers. These tools have built-in testing functionality. Ability to collaborate with existing tools is the main reason that works behind choosing testing tools in their company.

In.3 is using Selenium for their overall testing. They are also using jMeter for performance testing. Budget plays an important role as a factor for choosing testing tool in their company. So they are using free version and open source tools. According to them, it is also an important factor to decide how much automation in testing is needed in a project.

In.4 has different layers of testing for each project that they need to cover with different tools. They use JUnit for their unit testing but all the other tools depends on the type of project and type of framework they are using.

*Q.3 Which kind of development process are you using?*

*a. How do you do testing?*

*b. What is your perspective on doing manual testing following agile methodology?*

*Ans.* In.1 is using Agile(Kanban) method because it is convenient for their projects. They have retrospective meetings once in a while to ensure that they are following the right method for their projects. Their testing process differs from project to project. For new projects, they follow the same phases of analysis, development, QA and testing. So, when a task is in the development phase, the unit tests are written and after the development, the task is tested. For QA, they have code reviews and also checks whether the unit tests has covered the code properly or not. Then the user acceptance test is done by the product owner whether the functionality of the product is doing as per requirement. For some systems, if they have Mule, which is a platform to build an application network, then it will involve MUnit to test the integration. They are making a new testing strategy, but at this moment they are doing unit tests, some integration tests, acceptance tests and some end-to-end tests. Manual testing is important in areas where acceptance testing is involved. Manual testing is also important for their understanding when a new task is started and they need to do the testing manually to get the overall idea for deciding to switch to testing tools. Some projects are too small to set up large automated testing tool environment, then they use manual testing. They also have some systems which are very old, so only manual testing goes well in that environment. Setting automated testing tools environment also takes time and it depends on how much it has been set earlier.

In.2 has been doing their projects following the waterfall method for many years as their working policy is pretty traditional. But they are now trying to do agile wherever they can in their development process. They are doing automated testing whenever they build something. Their acceptance testing and release testing are done manually after the development phase. They involve customers to do the acceptance testing while providing them with some test cases, and later they get feedback from them. It is a challenging and time consuming effort to maintain the tests and software.

In.3 also have used waterfall method for their projects, but now they are strictly following agile methods. As soon as their development phase ends, manual testers start testing and at the same time they are running builds to ensure that the functionality of the software is not affected. So they are doing manual testing as well as using Selenium as their testing tool to run builds for overall testing of the software. Manual testing is as important as testing tools because not everything can be automated.

In.4 are following an agile(Kanban) approach for their products, but they are trying to switch to DevOps method for their development of software. The collaboration of development and testing team starts from the beginning of projects. According to In.4, manual testing is still very important. Some of the testing like regression testing should automated, but when it comes to acceptance testing, manual testing is appropriate.

*Q.4 How are you doing regression testing for your projects/products? Which tools you use for it?*

*Ans.* In.1 is using JUnit continuously to cover a lot of regression testing. Also their acceptance testing covers some regression testing manually. Currently, they do not have any big regression testing suite, but according to them, it would be great if they could have that, but these tools also take time to set up.

In.2 is not using any specific tools for regression testing. They run the test cases that has not been covered or show bugs. They collect feedback from their customers and then they look for the part where the error is showing in the source code and fix that.

In.3 is using Selenium for regression testing. They run it for several hours to find if the changes has caused any harm to the software and fix the erroneous parts.

In.4 is using several tools appropriate for each layer of their development phase for regression testing. They are using JUnit for unit testing. They also use a behavior-driven development tool called Cucumber.

*Q.5 How often do you do code coverage testing? Which tools do you use for it? How much satisfied are you with the tool's performance?*

*a. How important is it for your projects/products?*

*b. What is your perspective on input validation testing, domain testing?*

*Ans.* According to In.1, they are doing code coverage testing every night. They use Jenkins and Covertura for test coverage, and SonarQube to get a nice overall report of the test results. SonarQube also checks if the test coverage is high enough and it fails the Jenkins build if the percentage is too low. They are aiming for 80% code coverage and currently they have in 79%. In.1 thinks that they are getting quite good results from the tools they are using for their code coverage testing. The 20% that are left include issues that are not highly prioritized. They are getting their priorities covered with these tools. It is important for all their new projects that they do code coverage testing. They are not doing any separate input validation testing. In the acceptance testing their users test the system with a set of inputs whether they are valid or invalid. They do not do domain testing.

In.2 is currently trying to do code coverage testing in all of their projects. They were using manual testing for code coverage by looking for the functionality that the customers will be looking at. They were earlier following the waterfall method so they did not have anything to do with code coverage. They do not do input validation testing nor domain testing.

In.3 is not doing code coverage testing currently. Recently, they are trying to develop a framework and when the framework will be developed, they are hoping to start using code coverage tools.

In.4 uses code coverage tools in the unit layer and EPR layer. They use code coverage to ensure all the parts of the software has been tested and they aim for 80-90% code coverage.

*Q.6 What are the factors your company consider to choose tools for project/product?*

*a. How much satisfying performance do they give?*

*b. What are reasons that influence your testing efforts?*

*c. In which situations(GUI testing, usability testing etc.) manual testing overpowers testing tools? If you can elaborate on this.*

*Ans.* In.1 thinks choosing testing tools depend on the type of project. Price is also a factor. The tool should not be too much difficult to operate and set up. It will be great if the tool can be integrated with the existing tools. Example as, if it is a Jenkins plug-in or the tool can be run from Maven, that will make it easier. For testing efforts, they always want better quality for their products. They also want to find the bugs early before it goes into production. For older systems, they need to stick to the manual testing. Also for already finished products, it is difficult to set up an automated testing environment so manual testing is the only choice to fix something small in last minute.

In.2 thinks that choosing testing tools is based on customer's operating system choice, how easily the tool can integrate with the development tool kits, cost of the tools etc. If the tool is easy to be integrated with the existing ones and also costly, the company will choose that tool to save man hours. They are quite satisfied with the results their testing tools are delivering. According to In.2, they are following an agile method and doing sprints, so they are doing testing every two weeks. When they followed the waterfall method, the testing part was pushed to the end, and it led to disaster some of the times. They also had to stick to manual testing efforts back then, and when changes occurred in projects, they would rely on testing tools.

In.3 thinks choosing testing tools depends on the decision of whether one is ready to use a tool which is free, where one has to put a lot of effort to achieve the goal, or whether one is ready to spend some money on tools that will get the job done easily. They choose tools depending on the need, importance and type. They are quite satisfied with their choice of tools. Their choice of tools influences their testing efforts. If there is a small change in the project, they prefer to use manual testing. If there is a big change, they will go for the manual testing at first to be sure of the outcome and then they will involve testing tools. If there is a coverage of 75%, then one have to test the other 25% manually.

According to In.4, they have a list of factors to choose testing tools. Some of them are easy to use, fulfills the purpose of the project, secure to use, cost etc. The testing efforts depend on the type of project. If the project needs more rigorous testing, then they try to do it like that. Thus, it is context based. In.4 also thinks that manual testing is still important as human effort needs for critical thinking. One does not have enough time or resource for automating everything. Even if people use testing tools, tools still need maintenance.

*Q.7 What is the most challenging situation you had during testing?*

*a. What do you do when there is not enough time for thorough testing?*
*Ans.* In.1 once did a project where in every acceptance testing phase, the users find bug and the system was not functioning properly. They fixed that by developing better routine for writing unit tests. Most of the time they try the best they can do for the testing. It is not very thorough but it is enough for their customers.

In.2 thinks doing testing in a regular basis through out the process is quite challenging. Developers do not like testing that much being involved in the development, and testing gets pushed back at the end. This is challenging to get them do the testing through out the process more efficiently. In practice, In.2 does the testing of more prioritized requirements first, and then, if they have time, they will do the rest of the testing.

In.3 thinks the challenging part is the collaboration between the developer team and the QA team. They face challenges due to having both teams in different countries and the QA team gets much less time to do their work. It gets even more complicated when the deadline is near, and they do not have time to do thorough testing. In that situation, they do whatever they can with extra manpower to meet the deadline by doing enough testing in a short time.

In.4 thinks the challenging part is to get enough time to do everything they can for a project. Also maintaining testing tools can be challenging from time to time. It also depends on the team collaboration how they are approaching the challenges. In.4 also tells the testers that it is not possible to test everything that much thoroughly. One needs to take some risk based on that.

## 4.2 Response Collection From Questionnaire

We have made 5 sets of questionnaire sample links for CEO and other staffs of IT/software companies, non-IT/software companies and universities. Through email, we have sent out questionnaire links to the 185 organizations who have more than 20 employees. We have not got any responses so far from universities but collected responses from the rest of the sets of samples. In total we have got 12 responses from these four sets of samples. Each set has similar questions and based on the scope of the answers, type of company and job role, the responses are quite different in some questions. Since this thesis is based on the outcome of the specialization course project "use of automated testing in Norwegian companies" and it also had a questionnaire with more close-ended questions, we are going to use some interesting aspects of that project here.

Lets start with the **summary of the specialization course project** so we can take some aspects that are common in both the project and the thesis.
Investigation of the use of test automation in software quality assurance in Norwegian companies and organizations represents the use of different kinds of testing tools in software quality assurance in different types of norwegian companies and organizations. For the investigation part, a questionnaire has been made following agile methodology to collect response from different types of norwegian companies and organizations. It was put on LinkedIn, sent out as email etc to get responses. The quantity of responses was not adequate with respect to the amount of send outs because the response deadline of the questionnaire was short. Even with the few amount of responses, the answers of each questions of the questionnaire turned out to be quite diversified. Due to poor number of responses, the statistical analysis of the responses could not be done. All the findings and the analysis of the questionnaire are listed below.

**Table 4.1:** Research Methodology of former questionnaire

| | |
|---|---|
| **Strategy** | Survey Methodology |
| **Data generation method** | Questionnaire |
| **Followed method of developing questionnaire** | Agile Methodology |
| **Data Analysis** | Percentage calculation |

The questionnaire have 9 sections like testing process, organization, testing methods etc., each containing a number of questions related to the section header. There is also a decision making question about automated testing and manual testing. Demographic questions section also holds question about interested respondents who would like to do an interview. Testing method section is about which testing methods are used in the development of software products. Questions about the Dept/organization responsible for testing phase are put in the organization section. Development process section contains questions about which testing processes are used in the development of software products. Tools section is about the tools that are being used in the testing of the software product. Questions about the experience a company have using the testing tool is in experience section. Cost section is about the expensiveness of the automation testing tools and the manpower to make the tools work. Manual testing Section contains questions about manual testing is where one needs to write test suites for the software application to perform the test on them manually instead of using testing tools. Documentation section refers to the documentation of testing processes and tools.

A mix of 37 close-ended and open-ended questions has made the questionnaire a bit lengthy to answer. Though it was less time-consuming for the respondents because there were more close-ended questions than open-ended questions. This discrimination in the number of open-ended and close-ended questions did affect the overall outcome of the questionnaire. The lack of getting conclusive result does not only depend on the poor response rate but also depend on the abundance of close-ended questions where there is no chance to know why the respondents have chosen the respective answers. Here we are going to highlight the interesting and relevant questions from the questionnaire in the following table:

**Table 4.2:** Some interesting responses of the questionnaire

| No. | Section | Question | Answer |
|---|---|---|---|
| - | **-** | Select scope to answer(Typical Project, Typical product, Purchase/evaluation, Typical to company) | More than 50% voted for Typical Product. So all the answers are based on product development. |
| 1 | **Testing Method** | How are you doing QA? | More than 90% voted for Manual Testing. |
| 2 | | Do you have a validation/verification strategy? | More than 80 percent respondents are doing their validation/verification procedure through a configuration management system/issue-tracker/backlog. |
| 3 | | Do you trace test cases and test results to requirements and vice versa? | More than 40% companies trace test cases and test results to requirements and vice versa. Though almost 35% companies do not trace their test cases |

| | | | |
|---|---|---|---|
| | | | and test results to requirements and almost 20% respondents do not know whether their company use tracing or not. |
| 4 | **Development Process** | Which development process is most used in the development of the software product in your company? | Almost 90% companies follow agile development model for their software development process. |
| 5 | **Tools** | Are you using Automated testing tools for your product? | 77.8% companies are using testing tools for their products. There are still 22.2% companies who do not rely on testing tools yet. |
| 6 | **Documentation** | How do you manage test reports from automated testing? | 55.6% companies manage their test reports from automated testing just for internal use. While 33.3% companies manage them as part of deployment documentation. |
| 7 | **Demographic Questions** | What type of company are you currently in? | More than 60% respondents come from 'Software Development' companies while less than 30% comes from 'Consultancy'. |
| | | What is your job role? | More than 35% of the respondents are Software Developers. Options like 'Team Lead', 'System Architect' and 'Others' have got less than 30% and less than 25% respectively. |

Now lets start with the sample of the questionnaire for *CEO of IT/software companies* of this thesis. There are 4 responses collected from this particular sample. Here, we are highlighting the interesting responses. The second sample that we are going to analyze here is from the particular sample that has been sent out to *employees or staffs of IT/software companies*. We have got 6 responses from this sample and this is the sample with most response rate. The responses collected from both samples are given below:

**Table 4.3:** Selective responses from the samples "*CEO and staffs of IT/software companies*"

| No. | Section | Question | CEO of IT/software companies | staffs of IT/software companies |
|---|---|---|---|---|
| - | - | Select scope of answer | Most of the respondents have selected their scope of answers to product. So all of their responses are based on product. [See Appendix A, Figure A.1] | Starting with the scope of the answers of the respondents, we have found out that most of the respondents have answered the sample questionnaire based on their company. [See Appendix A, Figure A.9] |
| 1 | **Testing Method** | How are you doing QA(Quality Assurance) in software development? | Most of the respondents have answered that they are using automated testing tools sometimes for QA process. And one respondent also has selected the option of never using testing tools for QA. Despite of the availability and popularity of testing tools in the market, companies still do not rely completely on testing tools for QA in practice.[See Appendix A, Figure A.2] | We have noticed that the responses based on the respondent's QA techniques are quite intriguing. Majority of the respondents have opted out manual testing as well as automated testing for their QA process. It can mean that they are using both techniques simultaneously for their QA in software development.[See Appendix A, Figure A.10] |
| 2 | | How do you trace test cases and test results to requirements and vice versa? | Traceability is important to inspect whether the software requirements are met properly or whether redundant and unnecessary | The respondents have not given any clear answer based on this question. Most of the respondents have different answers |

| No. | Section | Question | CEO of IT/software companies | staffs of IT/software companies |
|---|---|---|---|---|
| | | | features are being developed in the software. The majority of the respondents of this particular set have chosen testing tools over other traceability techniques. Using testing tools for traceability does have it's perks of getting works done fast and bug free. [See Appendix A, Figure A.3] | and it makes it difficult to come to a decision about which technique is being mostly used. |
| 3 | **Tools** | How are you doing regression testing for your project/product? | There are contradictory responses from the respondents when there are questions like whether they are using testing tools and how they are doing regression testing. It is astonishing to know that majority of them are using testing tools but when it comes to regression testing, majority of them are relying on debugging. Using debugging for regression testing is a labor-some error-prone process, not to mention time consuming. [See Appendix A, Figure A.5] | The responses on using testing tools and techniques of doing regression testing have been contradictory. Though most of the respondents have given positive answers based on using testing tools, they have also voted for manual testing to do their regression testing. As it has been said earlier, using debugging as a technique for doing regression testing is enough set serious drawback in software testing phase. [See |

| No. | Section | Question | CEO of IT/software companies | staffs of IT/software companies |
|---|---|---|---|---|
| | | | | Appendix A, Figure A.12] |
| 4 | | Why are you doing regression testing using debugging following agile methodology? | To reveal the reason of using debugging for regression testing, the respondents have also given their opinion to the next question of reason they are using debugging for regression testing. The responses of this question highlights the reason of using debugging for regression testing following agile method is the project/product development time is short. It takes time to set up automated testing tool environment before executing any test with the tool. If the project/product development time is short, then it will be a waste of time to try to set up automated testing tool environment. It will be more efficient to the regression testing manually then.[See Appendix A, Figure A.6] | Unlike the former sample of the group *CEO of IT/software companies*, this time majority of the respondents have not given any clear response on why they are choosing debugging over testing tools while following agile method. As the majority of the respondents have given their responses based on their companies, it can be the company rule to use debugging for regression testing. [See Appendix A, Figure A.13] |

| No. | Section | Question | CEO of IT/software companies | staffs of IT/software companies |
|---|---|---|---|---|
| 5 | **Manual Testing** | What are the factors to choose Manual testing over Automated testing for the product development? | It has been seen from the responses that respondents are still caught up to manual testing rather than using testing tools. When they have been asked the reason behind this fact, their responses have been quite interesting. Majority of the respondents voted for factors like cost of the testing tools, developer's experience on handling testing manually or their lack of experience on certain tools and product type such as if there is usability, GUI testing etc involved or if the development stage is still early. [See Appendix A, Figure A.7] | The respondents have given some clear view about choosing manual testing over testing tools in general. It appears that cost of the tool, product development lifetime, developer's experience on handling testing manually or developer's lack of experience in using testing tools, product type involving usability testing etc. work as factors time to time to let the respondents choose manual testing.[See Appendix A, Figure A.14] |
| 6 | **Demographic Questions** | What is your job role? | The respondents also have confirmed their business area and job role in the sample questionnaire so that we can be assured to have responses from people experienced in fields related to the topic of our sample questionnaire. Among | We also have collected information about the respondent's expertise area and company type to ensure the relevance of their field of work to the topic of our questionnaire. Most of the respondents are team leads in this sample.[See |

| No. | Section | Question | CEO of IT/software companies | staffs of IT/software companies |
|---|---|---|---|---|
| | | | them, 50% are QA managers, others are team lead and testing manager. [See Appendix A, Figure A.8] | Appendix A, Figure A.15] |

We have got only one response each in both groups of *CEO and other employees of non-IT/software companies*. So we are going to highlight the interesting and relevant parts of them together.

Both of the respondents have chosen the scope of their answers based on product. Both of the parties use both manual and automated testing for QA in their software development. They both have some interesting response according to their regression testing techniques. The response from CEO group involves using only debugging and never using testing tools for regression testing. The respondent from the employee group does take into account both methods while doing regression testing. [See Appendix A, Figure A.16]

However, both of the respondents have given their side of reason of choosing manual testing for regression testing following agile method. The CEO group has selected the product lifetime as their reason for choosing debugging for regression testing. If the product developement time is short then investing time in building the testing tool environment is not efficient. So, in that case the regression testing is done by debugging because it is not that hard to keep up with a small project with small development time. But the option chosen by the employee group is not that clear. The option they have chosen could mean company policy. [See Appendix A, Figure A.17]

There is also a question about asking the factors of choosing manual testing over testing tools. Both groups have answered that question. One group has nominated cost of testing tools and product type such as products involving usability testing, GUI testing etc. as the reasons for choosing debugging over testing tools. And if the product is in early development stage, debugging comes handy to ensure the correct outcomes of testing. The other group has voted product development lifetime and product type such as products involving early development stage, usability testing, GUI testing etc. as the reasons for choosing debugging over testing tools. If the product development lifetime is short, it is inefficient to invest time in building testing tool environment. Choosing debugging in a project with short development lifetime can reduce all the hassle that comes up with the preparation of testing tool environment and save time. [See Appendix A, Figure A.18]

Both of the respondents belong to non-IT/software companies and both of them are software developers. So the respondents do not belong to IT/software companies but their job role is relevant to the area of the questionnaire.

If we compare Table 4.2 and Table 4.3, we notice some similarities in the questions. Although the questionnaire used in this thesis is quite different from the questionnaire of the specialization project. The questions in the present questionnaire are more open-ended than the questions in the questionnaire of specialization project. Even with the same questions, present questionnaire provide options that are able to give answers to the questions how much and why. We also need to remember that, the questionnaire of the specialization project had not been sent separately to different groups, so it contains a mixture of responses from CEO to staff, from IT to non-IT companies.

As we have created different sets of the sample questionnaire, we have been able to distinguish which group has chosen which options. While selecting the scope of the project, former respondents has chosen mostly product and present respondents have chosen product and company.

Both questionnaires contain questions relating QA. While Former respondents have chosen mostly debugging for QA, present respondents have some interesting answers. Due to open-ended questions, present respondents had the privilege to choose the frequency of using methods like manual testing, automated testing etc. for QA. So, the CEO group has confirmed using testing tools sometimes for QA and Staff group is using testing tools and debugging simultaneously for QA.

Unlike former questionnaire, present questionnaire has given us answers about the traceability techniques the respondents are using. But the former questionnaire has only been able to answer the user percentage of traceability in the respondents.

So it can be said that the present questionnaire is a substantial modification of the former questionnaire containing a good number of open-ended questions to know the 'how' and 'why' answers from the respondents.

# Chapter 5

## Data Evaluation and Discussion of Results

### 5.1 Discussion of results from Questionnaire

The difference of responses among the four sets of questionnaire samples are not only confined in numbers of participants. Their responses are different from IT to non-IT and from company to company. Even the job roles of the participants also impacts the nature of responses. Although survey method and questionnaire fall in the quantitative research method category, further analysis using quantitative research method needs a good response rate to statistically analyze them. Unfortunately, we could not get that much response. So, we are going to use qualitative research method to analyze further data. [52]We have used purposive sampling, which is one of the most known sampling methods where there is a preselected criteria relevant to selective research question. We already have four sets of samples of the questionnaire and relevant research questions. We have four criteria of sampling; CEO and staff groups of IT and non-IT companies. This sampling method becomes successful when data analysis goes along with the data collection.

At first, lets discuss the interesting responses between the two groups of IT companies and they are CEO and other staffs of IT companies. The respondents of the CEO group have answered the sample questionnaire based on product and company. And the respondents of the staffs group have answered the sample questionnaire based on project, product and company. Moreover, the respondents of CEO group are team lead, QA managers and testing manager. The respondents of staffs group have job roles like team lead and system engineer. So the answers of both of these groups are bound to be little different.

When it comes to doing QA in software development, the CEO group has decided to choose manual testing always and sometimes they use automated testing. But for the staffs group, they have chosen automated testing as well as manual testing for QA in their software development. They are using automated testing with a mix of manual effort to do QA in their software development.

Respondents from both of the groups have agreed on using testing tools for their product/project.

Respondents from both of the groups are following regression testing for their product/project. But the respondents of CEO group are always doing manual testing and sometimes automated testing for their regression testing. On the other hand, the respondents of staffs group are using automated testing with a mix of manual effort for regression testing.

Respondents of both groups prioritize manual testing over automated testing when their project/product development time is short. But for the respondents of CEO group, they are always doing regression testing manually because usually they are handling product/project which have short development lifetime.

There are times when situations are in the favor of using manual testing for product development. Respondents of both groups have agreed to the facts like cost of the tools, product development lifetime, developer's experience, product type(early stage of development, usability testing, GUI etc.) etc to favor annual testing for product development.

While discussing the responses we have got from CEO and staffs groups of non-IT companies, it is important to note that we only have got one response per group. Respondents of both groups have answered the questionnaire sample based on product. But as they are from different companies with different job roles, some answers might be different in this context.

According to the respondent of CEO group, they are using only manual testing for regression testing of their project/product without the help of testing tools at all. On the contrary, the respondent of the staffs group are collaborating manual effort with testing tools for regression testing of their project/product.

Choosing debugging for regression testing of project/product is a controversial fact. The respondent of CEO group has answered this by choosing short development time of product/project as a reason. But the respondent of staffs group has chosen something discreet as 'Other'. It means the reason may involve company policy or rule of not using testing tools for regression testing of their product/project.

There are some situations when choosing manual testing over automated testing for product development is wise. According to the respondent of CEO group cost of testing tools and products that involves usability testing, GUI testing etc. are the factors for favoring manual testing over testing tools. But instead of cost of the tools, the respondent of staffs group has gone for product development lifetime as a factor for this. So, it may mean that, the respondent of staffs group belongs to a large-scale well funded company who do not count the cost of tools as a major fact.

The discussions of these four groups or in this case, IT and non-IT group create some interesting findings.

The respondents of IT group are using testing tools at some point in their regression testing more or less while the respondents of non-IT group are less likely to do that. Being in an IT company does give the privilege to know more about new technicality and their practice in real life rather than an IT department of a non-IT company. Using testing tools does require technical knowledge and a good hand in programming.

We did not get expected number of responses from the questionnaire samples. Though our questionnaire is based on the previously done questionnaire for the specialization project course, it has a complete new set of open-ended questions with different purpose than the previous one(See Appendix C). Also we have sent out to 185 companies both IT and non-IT companies who have more than 20 employees. But it has proven quite hard to get much responses from the companies and after out multiple tries we have got only 12 responses. It is quite less than the number of senders but we have managed to get some interesting, different answers from the respondents. So, to make up the void of the questionnaire's low response rate, we have done four interviews where each interviewee is from each criteria.

## 5.2 Discussion of the answers from Interview

After getting small amount of answers from he questionnaire sample, we decided to do interview with the respondents. Five respondents from four sets of questionnaire sample have answered positively for the idea of interview and four of them have been able to give their interviews. The interview questions aims to find the answers related to research questions. It includes questions about development process, testing method, some point of view from the interviewees etc(See Appendix B). Each respondent belongs to each set of the questionnaire sample. Their answers have varied from question to question based on their experience, company type, company policy, job role etc.. Lets discuss their answers according to the topics of the questions.

**Table 5.1:** Evaluation of four interviews

| No. | Key Point | Explanation |
|---|---|---|
| 1 | Software development process | All the interviewees are currently using agile method or a type of agile method(Kanban) for their software development. But, In.1 still has to use waterfall method for their traditional old systems which has not been integrated with the new digital system yet. Also In.2 and In.3 had been using waterfall method for a long time because of their traditional old system which was not compatible to follow agile |

|  |  |  |
|---|---|---|
|  |  | method. In.4 is the only one have been following agile(Kanban) method for a very long time and currently they are trying to convert their software development process into DevOps for better customer satisfaction and efficient product delivery. The overall scenario is quite interesting. While two companies are clearly keeping themselves updated by following one of the most popular development processes, even deciding to go for new development process like DevOps, other two companies are still struggling to migrate their software development process from waterfall method to agile method. |
| 2 | QA process | According to In.1 and In.4, they are following the typical analysis, development, QA and testing paradigm. They are doing analysis of a requirement, break it into small tasks, develop them and check them for QA and testing in every cycle. For these two interviewees, development and QA run in parallel for every task along with testing. Same goes with In.2, but in a different way as they are following PRINCE2, a process based approach to project management. Different scenario is seen in the case of In.3 as their developement team and QA team are not in the same country. So, they face probelms as QA gets pushed back because of long development time and lack of collaboration of work between two teams during the development phase. They have to hurry their QA and testing part to meet the deadlines. |
| 3 | Testing tools | According to In.1 and In.4, they are both using JUnit for their usit testing. For system integration and code coverage, In.1 is using Jenkins and SonarQube while In.4 is using a number of different tools in their different layers of testing fulfilling different purpose. In.3 is using Selenium for all the testing they are doing and for performance testing they are relying on jMeter. In.2 is primarily using Team Foundation server in the building process and other build servers. Almost all the interviewees agree on the reason of the popularity of testing tools in their companies is getting better products as they are well tested with the help of the tools. The testing takes less time than manual and there is far less chance of jeopardizing the source codes while using tools for testing. |

| | | |
|---|---|---|
| 4 | Manual testing in agile method | Although all interviwees are using testing tools for different phase of software development, manual testing is still important for them. Acceptance testing is done manually. And for new systems or small changes in the project, at first they prefer manual testing to know about the outcome. Also they still have some pretty old part of the system which can only be tested manually. In.2 finds it challenging and time consuming effort to maintain the tests and software. According to In.3, not everything can be automated and it is also an important factor to decide how much automation in testing is needed in a project. So it seems no matter how much popular testing tools get, manual testing will still be a major part for companies to do their testing for software development from time to time. |
| 5 | Regression testing | In.1 is using JUnit for their regression testing. In.3 is using Selenium for regression testing. In.4 is using different tools for regression testing in different levels. They are using JUnit for low levels. They also use a behavior-driven development tool called Cucumber. It is a bit strange that In.2 is not using any specific tools for their regression testing. They are saying the integrated tools in visual studio do the checking of functionality and when it finds a bug, they fix it. It seems all of them are using different tools as per their needs for regression testing. Some of the companies like the one of In.4 need rigorous regression, so they are applying different tools for different levels and some of companies like the one of In.2 do not prioritize it. |
| 6 | Code coverage testing | It is interesting to know that only In.1 and In.4 are using some sort of code coverage tools. In.1 is using Jenkins and Covertura for test coverage and SonarQube to get a nice overall report on the test results. SonarQube also checks if the test coverage is high enough and it fails the Jenkins build if the percentage is too low. In.4 is using code coverage tools in unit layer and EPR layer. Currently, In.2 is trying to do code coverage manually and In.3 is hoping to start it doing in near future. Code coverage testing ensures the functionality of product and makes it ready to deliver. Doing code coverage manually seems a laborous work not to mention time consuming and it does not gurantee a an error free result. Not doing code coverage makes it even worse. |

|  |  |  |
|---|---|---|
| 7 | Reasons for choosing testing tools and their performance | According to all of the interviewees, they are quite content with the performance of their current testing tools. They also have mentioned the reason behind choosing these tools. Each has their own perspective and opinions based on company policy for choosing tools. But all of them have given some similar points like cost of the tools, need, importance and type of the tools etc. In.4 has been the only one emphasizing about the tools being secured due to their course of projects. Basically if there is a free tool which is easy to use, can be adapted in the existing system or tool and fulfills the purpose, any company would go for it. But if the tool is costly but still offers all the advantages, then some company would buy it and some would not due to budget reason. |
| 7 | Challanging experience | The interesting challenging experiences are described by  In.2 and In.3. For In.2, doing testing in a regular basis throughout the process is quite challenging. Developers do not like testing that much being involved in the development. And testing gets pushed back at the end. So this is challenging to get them do the testing throughout the process more efficiently. When there is less time for testing, In.2 does the testing of more prioritized requirements first and then if they have time, they will do the rest of the testing. According to In.3, collaborating the developer team with the QA team is challenging. They face challenges due to having both teams in different countries and the QA team gets much less time to do their work. It gets even complicated when the deadline is near and they do not have time to do the thorough testing. In that situation they do whatever they can with extra manpower to meet the deadline by doing enough testing in a short time. It seems pretty obvious that lack of collaboration among teams can create disaster for a project. Getting less time for testing and lack of collaboration between developer team and QA team can yield an erroneous end product which will get bad review and face loss. |

## 5.3 Linking the results from Questionnaire and Interview

Some of the similarities and differences between the responses from questionnaire samples and the answers of the interview are given below:

**Table 5.2:** Similarities and differences between the responses from questionnaire samples and interviews

| No. | Key Points | Explanations | |
|---|---|---|---|
| 1 | Regression testing | Differences | The responses from questionnaire samples show more supports for doing regression testing manually. Though there are a lot of responses indicating using automated testing for regression testing mostly, but they have voted for manual testing as well. From the answers of the interview, three of them are using various types of testing tools for their regression testing. This is a clear difference in opinion from both sides. |
| | | Similarities | So this issue can be solved by pointing out the responses from the question asking about the reason of doing regression testing manually in the questionnaire. Most of the respondents have agreed on the point of using manual testing for regression testing because of short development time of project/product. So it seems they will choose manual testing when there is an exception. |
| 2 | Manual testing | Similarities: According to the most of the respondents of the questionnaire samples, the factors that will lead them to choose manual testing over testing tools are cost, product development life time, developer's experience and product type where it is early stage of development and involves GUI testing, usability testing etc.. According to the answers from the interviewees, we get that they will also choose debugging over testing tools in similar | |

| | | |
|---|---|---|
| | | situations. In addition, they also seem pretty confident about using manual testing time to time whenever it is needed in their projects. According to interviewees, not everything can be automated and there is no need to do that. |

So, the findings so far pretty much answers the research questions we have. Based on the findings of my specialization project, there are two research questions. Lets discuss them:

*RQ1.* Why the usage of testing tools are still not that popular in the software industry in Norway?

Ans. From the analysis and discussions of the responses from questionnaire and answers from interviews, it seems that the use of testing tool is popular in the software industry in Norway. But it is still not that much popular despite the advantages it offers and the warm approvals it get from literature review. In practice, companies still are using manual testing time to time. Sometimes it is because of the exception like short development life time of the project and sometimes because of the tradition or policy of the company. According to some companies, testing tools take so much time for preparation and maintenance and some also think tools are quite complicated for day to day navigation. Such as, if a company is traditional and still follows the waterfall method, then using testing tools can not be easy for that company. So, that company is bound to pick monotonous debugging over time-savvy testing tools. Some also think that there are excellent testing tools available with higher price and very few companies allow to buy such expensive testing tools. So, sometimes it is the situation or sometimes it is the features of testing tool itself lead people towards manual testing.

*RQ2.* How is it going on doing manual testing while following agile methodology for developing software/project?

Ans. Some companies are still doing manual testing while following agile method because not all of their system is modernized. Some part of their system is still pretty old so they are using manual testing for those parts. Even in agile method, everyone is using manual testing for acceptance testing, GUI testing etc. These types of testings are done best manually from a human's point of view. And also when the a new task is developed, at first developers try to test it manually to be sure of the results and then they write test cases for that part to do the testing with tools. Again when there is a small modification in the system, it is wise to check it manually rather than running a long build for that small change with testing tools. According to the interviewees, if there is a coverage of 75%, then one still have to test the other 25% manually. So it seems like not a problem to use debugging while following agile method when there are situations like these. But for regression testing, if one tries to do it manually, then it is not wise. It is all about balancing the use of testing tools with manual efforts from time to time.

Lastly, we can say that people use manual testing; sometimes because they are bound to and sometimes because it is easier than using testing tools for that particular scenario. We agree to the fact that the results we have obtained so far are not conclusive. The response rate of the questionnaire sample was quite low. We repeated the survey method even after specialization project because this time, we have been able to develop a more precise, open-ended questions included questionnaire. But, it has been proved to us that, no matter how much attractive and precise the questionnaire is, the response rate does not go high. We did start to send the sample a bit late than expected but these 185 companies are those we were aiming at. We had time to resend the sample through email and we did that. Nothing seemed to have raise the response rate of the sample. It seems, we have got more conclusive and interesting answers through interviews rather than the questionnaire. After this experience, we do not think questionnaire is a well-thought technique to collect data as achieving higher response rate through it, is very tricky. There should be more types of effective techniques like interview for collecting data so that researchers can get decent response rate and can do a conclusive data analysis without letting their hard work go in vain.

# Chapter 6

## Conclusion

The use of testing tools for software is inevitable in Norwegian companies. But, we also can not forget the contribution of manual testing either. It does not mean that the importance of using testing tools is lacking in these companies. Rather it motivates the balancing act of using testing tools and debugging. Combining the use of testing tools with proper of manual effort is going to make the software testing phase more efficient and cost-effective in terms of labor and money. For future reference, a more precise and effective data collection method like interview can be developed to collect a wide range of data efficiently. With that method, one can analyze and determine the real scenario of the use of testing tools in Europe or even worldwide.

# References

**[1]** T. Maneela and D. Gaurav, “A Research Study on importance of Testing and Quality Assurance in Software Development Life Cycle (SDLC) Models", International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-2, Issue-3, July 2012

**[2]** Specialization project on "Investigation of the use of test automation in software quality assurance in Norwegian companies and organizations" by Saraf Anika, Carl-Fredrik Sørensen, NTNU, 2015

**[3]** N. Ana Candida C., R. Ana Regina C. da, T. Guilherme H., M. Paula G., Integrating Verification and Validation Techniques Knowledge into Software Engineering Environments

**[4]** Heitmeyer C. L., Formal Methods for Specifying, Validating, and Verifying Requirements, Naval Research Laboratory, Washington, DC, 2007

**[5]** Easterbrook S., Callahan J., Formal Methods for Verification and Validation of partial specifications: A Case Study, NASA/WVU Software Research Lab, NASA Independent Verification and Validation Facility, USA, 1997

**[6]** Saswat Anand et al., An Orchestrated Survey on Automated Software Test Case Generation, Antonia Bertolino, J. Jenny Li and Hong Zhu (Editor/Orchestrators), Journal of Systems and Software, 2013

**[7]** mapwrite.com.au. http://www.mapwright.com.au/quality_manual.html

**[8]** iso.org. ISO 9000:2005, Clause 3.2.11

**[9]** E. Emad Kh., A. Ibrahim M., New Model to Achieve Software Quality Assurance (SQA) in Web Application, July 2012

**[10]** Bacchelli A., Bird C., Expectations, Outcomes, and Challenges Of Modern Code Review, Proceedings of the 2013 International Conference on Software Engineering(ICSE '13 ), Pages 712-721

**[11]** Fagan M.E., "Design and code inspections to reduce errors in program development," IBM Systems Journal, 1976

**[12]** Shull, "Inspecting the History of Inspections," 2008

**[13]** W. Laurie, K. Gunnar, N. Nachiappan, On the Effectiveness of Unit Test Automation at Microsoft, 2009

**[14]** T. Hideo, P. Mukul R., R. Sreeranga P., F. Masahiro, Automated System Testing of Dynamic Web Applications, 2011

**[15]** P. Christer, Y. Nur, Establishment of Automated Regression Testing at ABB:Industrial

Experience Report on ‘Avoiding the Pitfalls’, Proceedings of the 19th International Conference on Automated Software Engineering (ASE’04)

**[16]** M. Ali, R. P. Mukul, Automated Cross-Browser Compatibility Testing, ICSE'11

**[17]** testingbasicinterviewquestions.blogspot.no. http://testingbasicinterviewquestions.blogspot.no/2015/02/functional-testing-vs-non-functional-testing.html

**[18]** T. W. Au Fiora, B. Simon, W. Ian, D. Gillian, Automated Usability Testing Framework, 2008

**[19]** M. Riyadh, E. Naeem, K. Thabet, M. Nariman, M. Sam, S. Angelos, A Whitebox Approach for Automated Security Testing of Android Applications on the Cloud, 2012

**[20]** guru99.com. http://www.guru99.com/static-dynamic-testing.html

**[21]** G. Patrice, L. Micheal Y., M. David, Automated Whitebox Fuzz Testing

**[22]** M. Leonardo, P. Mauro, R. Oliviero, S. Mauro, AutoBlackTest: Automatic Black-Box Testing of Interactive Applications, April 2012

**[23]** D. Brett, D. Danny, G. Kely, M. Darko, Automated Testing of Refactoring Engines, ESEC/FSE’07, September 3–7, 2007

**[24]** multisoftsystems.com. http://www.multisoftsystems.com/blog/why-you-should-move-from-manual-testing-to-automation-testing/

**[25]** U. Nidhika, C. Vinay, Design and Implementation in Selenium IDE with Web Driver, May 2012

**[26]** A. Samer, E. Derar, G. Joy, Automated User Interface Testing for Web Applications and TestComplete, 2012

**[27]** D. Hyunsook, R. Gregg, K. Alex, Empirical Studies of Test Case Prioritization in a JUnit Testing Environment, 2004

**[28]** testingtools.com. http://www.testingtools.com/test-automation/

**[29]** Alemerien K., Magel K., Examining the Effectiveness of Testing Coverage Tools: An Empirical Study, International Journal of Software Engineering and Its Applications Vol.8, No.5 (2014), pp.139-162

**[30]** Mockus A., Nagappan N., T. Dinh-Trong T., Test coverage and post-verification defects: A multiple case study, ESEM '09 Proceedings of the 2009 3rd International Symposium on Empirical Software Engineering and Measurement, Pages 291-301

**[31]** chambers.com.au. http://www.chambers.com.au/glossary/traceability_analysis.php

**[32]** Gotel O.C.Z. and Finkelstein A.C.W., An Analysis of the Requirements Traceability Problem, Proc. Of the First International Conference on Requirements Engineering, Colorado Springs, 1994:

94-101

**[33]** Khursheed F., Suaib M., A Survey on Importance of Requirement Traceability in Software Engineering, International Journal of Engineering and Innovative Technology (IJEIT) Volume 5, Issue 4, October 2015

**[34]** ldra.com. http://www.ldra.com/en/tbreq

**[35]** istqbexamcertification.com. http://istqbexamcertification.com/what-is-agile-model-advantages-disadvantages-and-when-to-use-it/

**[36]** Begel A., Nagappan N., Usage and Perceptions of Agile Software Development in an Industrial Context: An Exploratory Study, First International Symposium on Empirical Software Engineering and Metrics, Microsoft Research, 2007

**[37]** Royce, W. W., "Managing the Development of Large Software Systems", Proceedings of IEEE WESCON 26 (August 1970): 1–9

**[38]** umsl.edu. http://www.umsl.edu/~hugheyd/is6840/waterfall.html

**[39]** Peterson K., Wohlin C., Baca D., The Waterfall Model in Large-Scale Development, 10th International Conference, PROFES 2009, Oulu, Finland, June 15-17, pp 386-400

**[40]** Debois P., "Agile 2008 Toronto", Just Enough Documented Information, Retrieved 12 March 2015

**[41]** Debois P., *"DevOpsDays Ghent"*, DevopsDays(2009), Retrieved 31 March 2011

**[42]** istqbexamcertification.com. http://istqbexamcertification.com/what-is-spiral-model-advantages-disadvantages-and-when-to-use-it/

**[43]** agiledata.org. http://agiledata.org/essays/tdd.html

**[44]** Bulajic A., Sambasivam S., Overview of the Test Driven Development Research Projects and Experiments, Proceedings of Informing Science & IT Education Conference (InSITE) 2012

**[45]** computerworld.com. http://www.computerworld.com/article/2585634/app-development/extreme-programming.html

**[46]** extremeprogramming.org. http://www.extremeprogramming.org/

**[47]** testingreferences.com. http://www.testingreferences.com/testinghistory.php

**[48]** K. Yuji, K. Kenji and H. Miyuki, Sania: Syntactic and Semantic Analysis for Automated Testing against SQL Injection. Keio University, 23rd Annual Computer Security Applications Conference, 2007

**[49]** C. George, B. Stefan, Z. Cristian, Automated Software Testing as a Service, Proceedings of the 1st ACM symposium on Cloud computing, SoCC'10, June 10–11, 2010

**[50]** B. Sebastian, E. J. V. Tanja, GUITest: A Java Library for Fully Automated GUI Robustness Testing, ASE'12, September 3–7, 2012

**[51]** M. R. Dudekula, R. K. M. Katam, P. Kai, V. M. Mika, Benefits and Limitations of Automated Software Testing: Systematic Literature Review and Practitioner Survey, 2012 IEEE

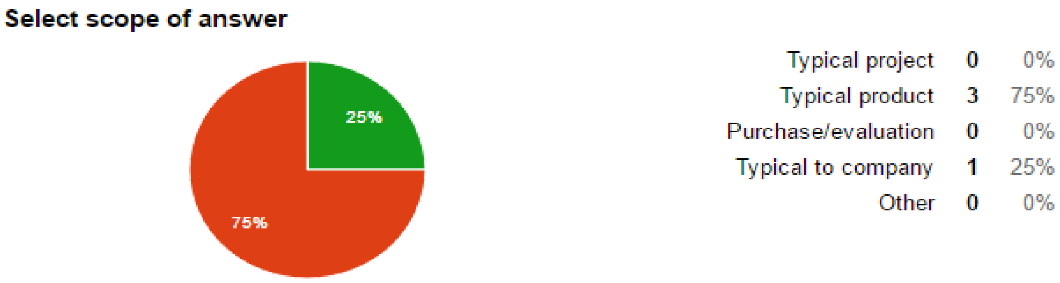
**[52]** Mack N., Woodsong C.,Macqueen K.M., Namey E., Qualitative Research Methods: A Data Collector's Field Guide, 2005, 1-13

# Appendices

## Appendix A : Response charts from Questionnaire samples in percentage

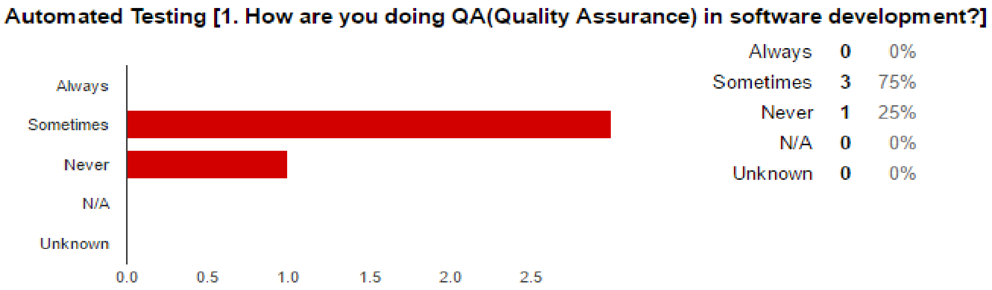
The selective interesting response charts(in percentage) have been obtained from the responses of questionnaire samples.

- **IT/Software Companies(CEO group):**



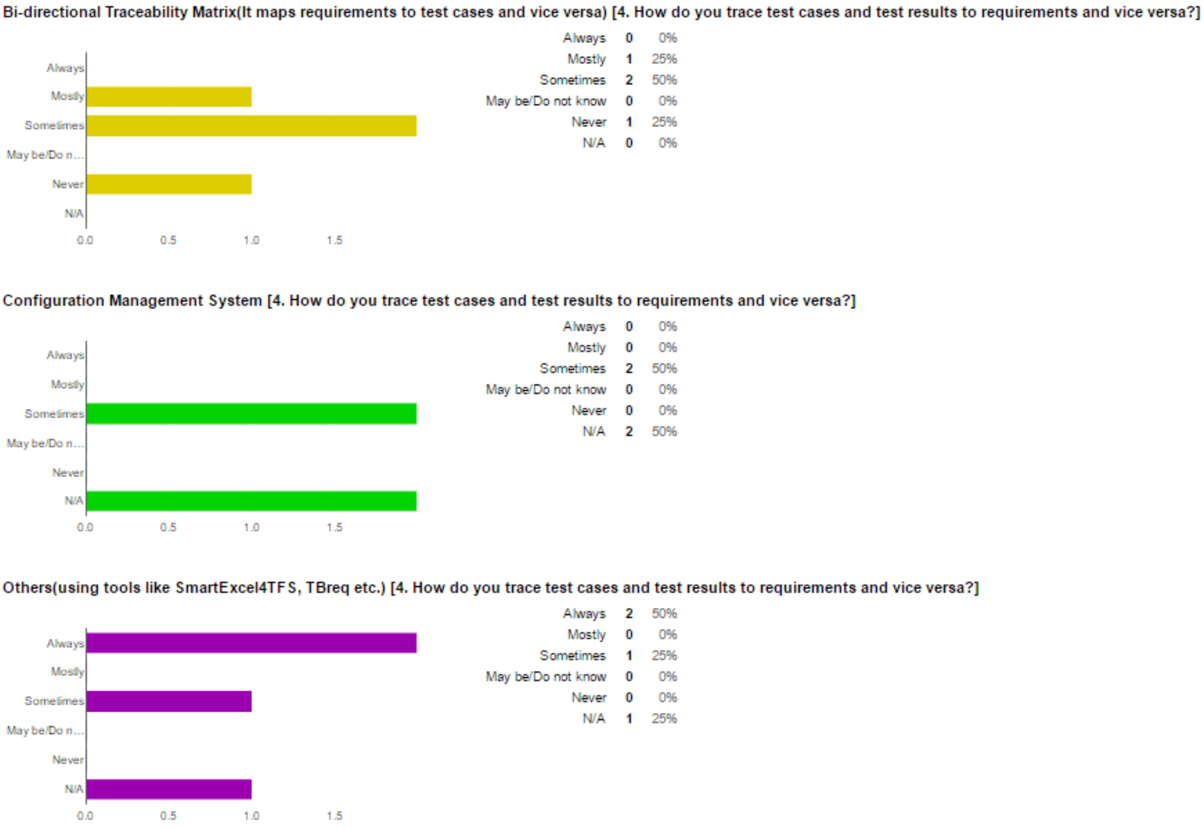
| | | |
|---|---|---|
| Typical project | 0 | 0% |
| Typical product | 3 | 75% |
| Purchase/evaluation | 0 | 0% |
| Typical to company | 1 | 25% |
| Other | 0 | 0% |

**Figure A.1:** Response analysis of the question “Select scope of answer”



| | | |
|---|---|---|
| Always | 0 | 0% |
| Sometimes | 3 | 75% |
| Never | 1 | 25% |
| N/A | 0 | 0% |
| Unknown | 0 | 0% |

**Figure A.2:** Response analysis of the question “1. How are you doing QA(Quality Assurance) in software development?”



Bi-directional Traceability Matrix(It maps requirements to test cases and vice versa) [4. How do you trace test cases and test results to requirements and vice versa?]

| | | |
|---|---|---|
| Always | 0 | 0% |
| Mostly | 1 | 25% |
| Sometimes | 2 | 50% |
| May be/Do not know | 0 | 0% |
| Never | 1 | 25% |
| N/A | 0 | 0% |

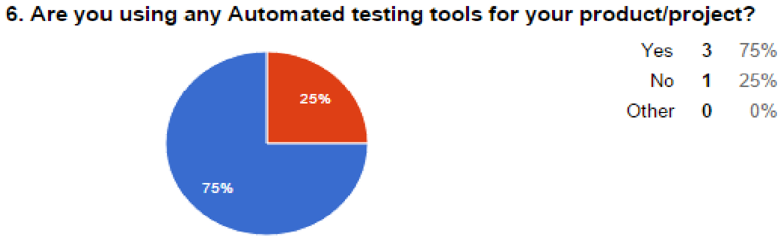
Configuration Management System [4. How do you trace test cases and test results to requirements and vice versa?]

| | | |
|---|---|---|
| Always | 0 | 0% |
| Mostly | 0 | 0% |
| Sometimes | 2 | 50% |
| May be/Do not know | 0 | 0% |
| Never | 0 | 0% |
| N/A | 2 | 50% |

Others(using tools like SmartExcel4TFS, TBreq etc.) [4. How do you trace test cases and test results to requirements and vice versa?]

| | | |
|---|---|---|
| Always | 2 | 50% |
| Mostly | 0 | 0% |
| Sometimes | 1 | 25% |
| May be/Do not know | 0 | 0% |
| Never | 0 | 0% |
| N/A | 1 | 25% |

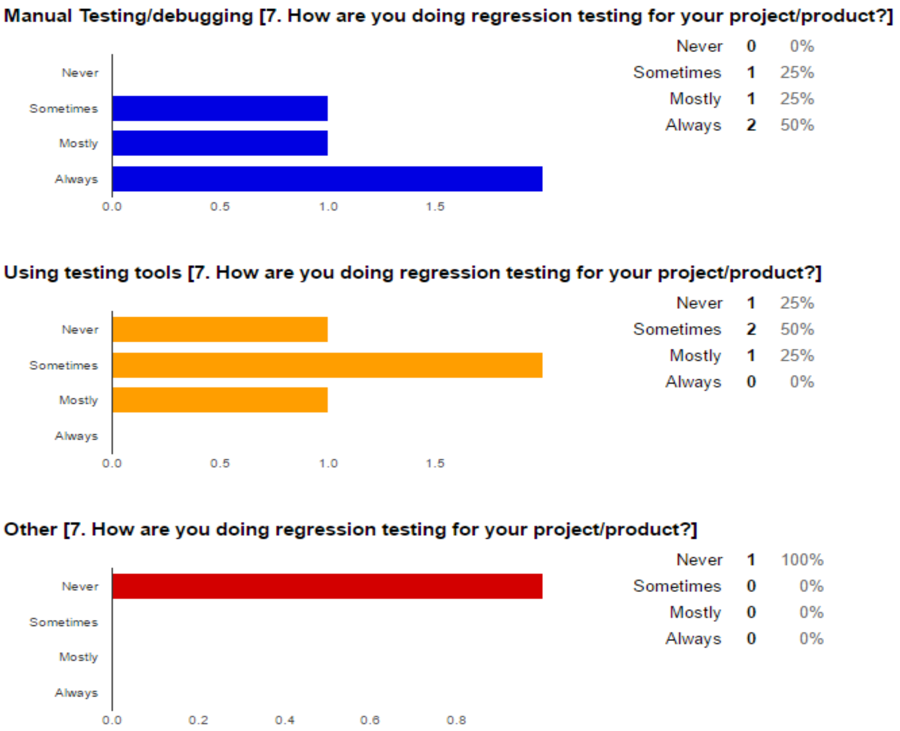
**Figure A.3:** Response analysis of the question “4. How do you trace test cases and test results to requirements and vice versa?”



6. Are you using any Automated testing tools for your product/project?

| | | |
|---|---|---|
| Yes | 3 | 75% |
| No | 1 | 25% |
| Other | 0 | 0% |

**Figure A.4:** Response analysis of the question “6. Are you using any Automated testing tools for your product/project?"


Manual Testing/debugging [7. How are you doing regression testing for your project/product?]

| | | |
|---|---|---|
| Never | 0 | 0% |
| Sometimes | 1 | 25% |
| Mostly | 1 | 25% |
| Always | 2 | 50% |

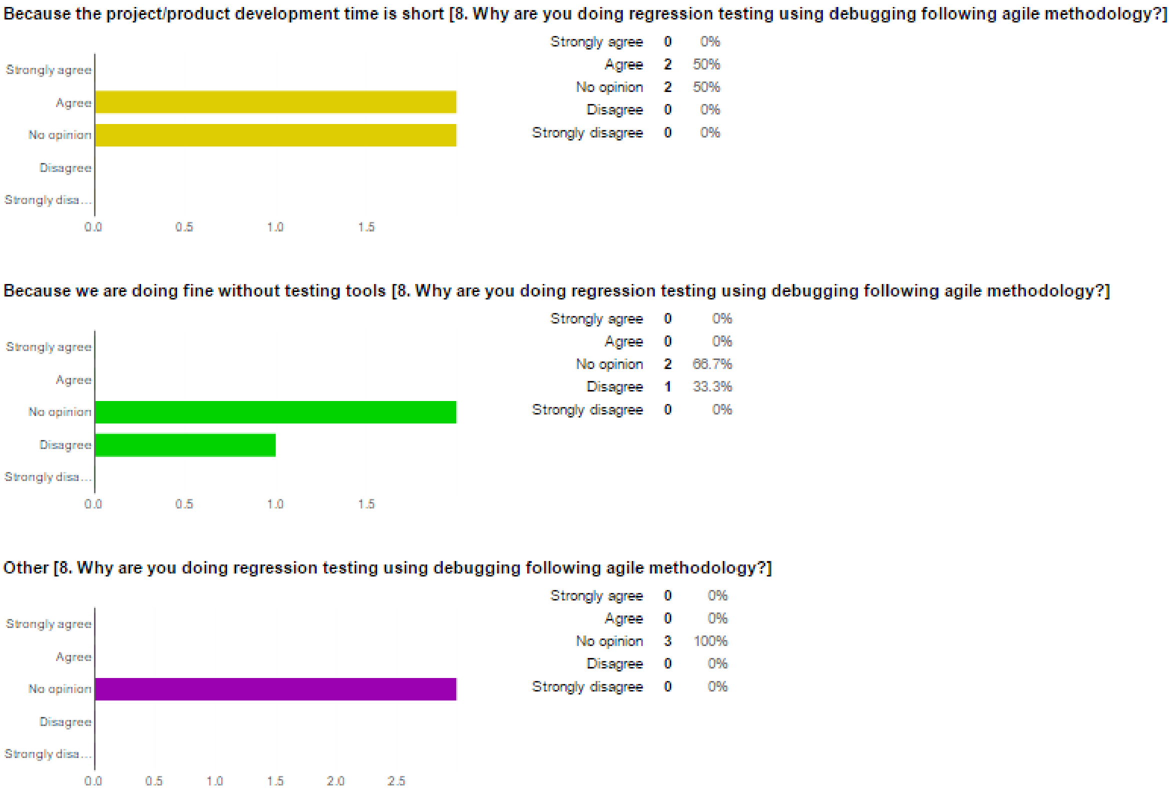
Using testing tools [7. How are you doing regression testing for your project/product?]

| | | |
|---|---|---|
| Never | 1 | 25% |
| Sometimes | 2 | 50% |
| Mostly | 1 | 25% |
| Always | 0 | 0% |

Other [7. How are you doing regression testing for your project/product?]

| | | |
|---|---|---|
| Never | 1 | 100% |
| Sometimes | 0 | 0% |
| Mostly | 0 | 0% |
| Always | 0 | 0% |

**Figure A.5:** Response analysis of the question "7. How are you doing regression testing for your project/product?”


Because the project/product development time is short [8. Why are you doing regression testing using debugging following agile methodology?]

| | | |
|---|---|---|
| Strongly agree | 0 | 0% |
| Agree | 2 | 50% |
| No opinion | 2 | 50% |
| Disagree | 0 | 0% |
| Strongly disagree | 0 | 0% |

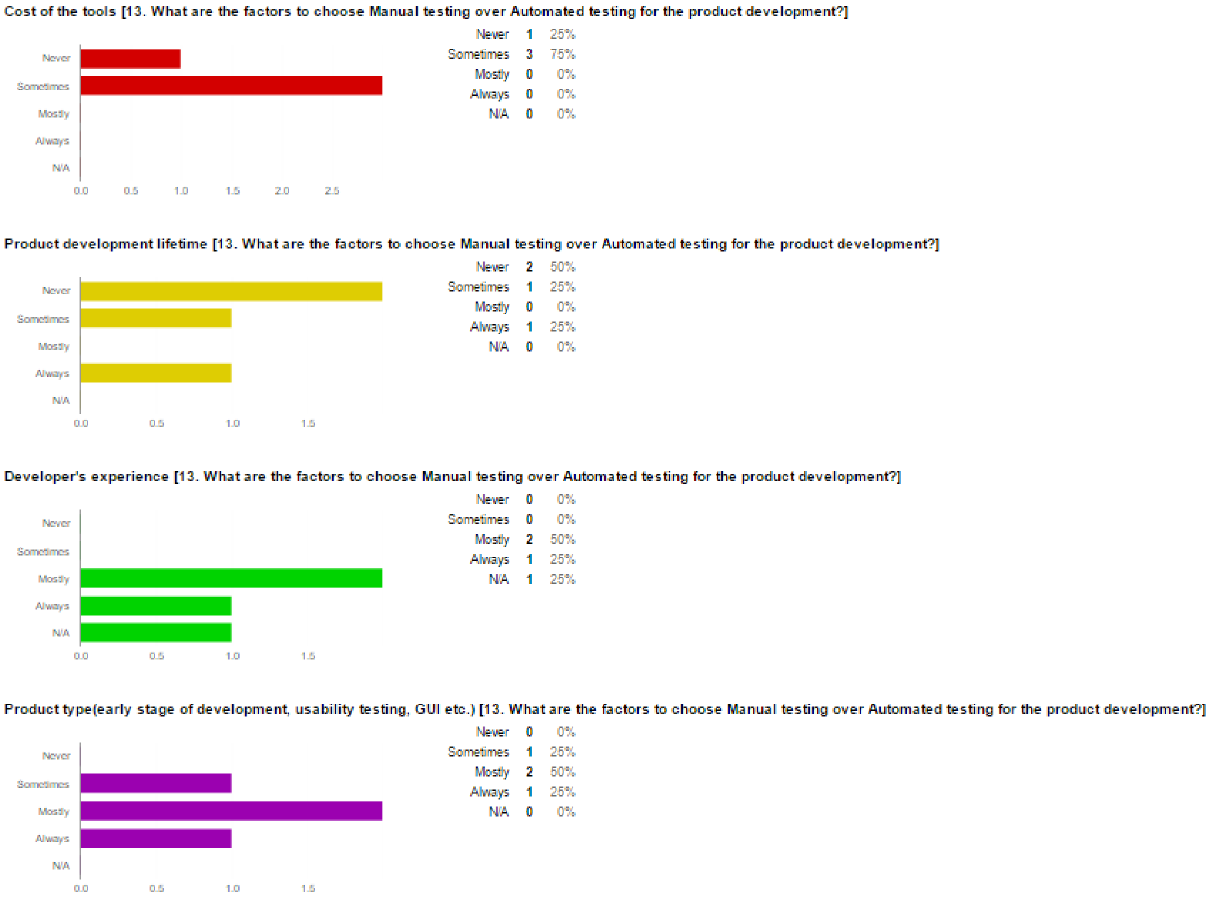
Because we are doing fine without testing tools [8. Why are you doing regression testing using debugging following agile methodology?]

| | | |
|---|---|---|
| Strongly agree | 0 | 0% |
| Agree | 0 | 0% |
| No opinion | 2 | 66.7% |
| Disagree | 1 | 33.3% |
| Strongly disagree | 0 | 0% |

Other [8. Why are you doing regression testing using debugging following agile methodology?]

| | | |
|---|---|---|
| Strongly agree | 0 | 0% |
| Agree | 0 | 0% |
| No opinion | 3 | 100% |
| Disagree | 0 | 0% |
| Strongly disagree | 0 | 0% |

**Figure A.6:** Response analysis of the question “8. Why are you doing regression testing using debugging following agile methodology?”



Cost of the tools [13. What are the factors to choose Manual testing over Automated testing for the product development?]

| | | |
|---|---|---|
| Never | 1 | 25% |
| Sometimes | 3 | 75% |
| Mostly | 0 | 0% |
| Always | 0 | 0% |
| N/A | 0 | 0% |

Product development lifetime [13. What are the factors to choose Manual testing over Automated testing for the product development?]

| | | |
|---|---|---|
| Never | 2 | 50% |
| Sometimes | 1 | 25% |
| Mostly | 0 | 0% |
| Always | 1 | 25% |
| N/A | 0 | 0% |

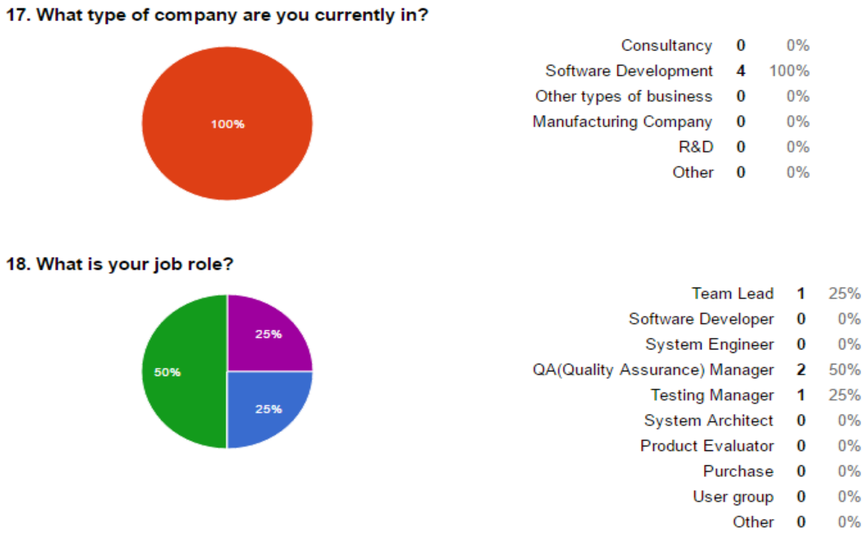
Developer's experience [13. What are the factors to choose Manual testing over Automated testing for the product development?]

| | | |
|---|---|---|
| Never | 0 | 0% |
| Sometimes | 0 | 0% |
| Mostly | 2 | 50% |
| Always | 1 | 25% |
| N/A | 1 | 25% |

Product type(early stage of development, usability testing, GUI etc.) [13. What are the factors to choose Manual testing over Automated testing for the product development?]

| | | |
|---|---|---|
| Never | 0 | 0% |
| Sometimes | 1 | 25% |
| Mostly | 2 | 50% |
| Always | 1 | 25% |
| N/A | 0 | 0% |

**Figure A.7:** Response analysis of the question "13. What are the factors to choose Manual testing over Automated testing for the product development?"



17. What type of company are you currently in?

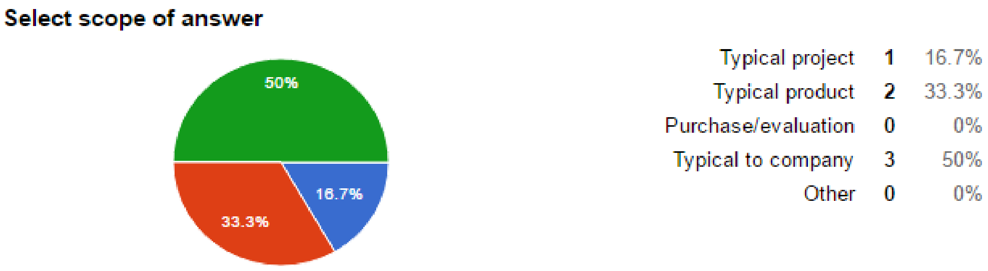
| | | |
|---|---|---|
| Consultancy | 0 | 0% |
| Software Development | 4 | 100% |
| Other types of business | 0 | 0% |
| Manufacturing Company | 0 | 0% |
| R&D | 0 | 0% |
| Other | 0 | 0% |

18. What is your job role?

| | | |
|---|---|---|
| Team Lead | 1 | 25% |
| Software Developer | 0 | 0% |
| System Engineer | 0 | 0% |
| QA(Quality Assurance) Manager | 2 | 50% |
| Testing Manager | 1 | 25% |
| System Architect | 0 | 0% |
| Product Evaluator | 0 | 0% |
| Purchase | 0 | 0% |
| User group | 0 | 0% |
| Other | 0 | 0% |

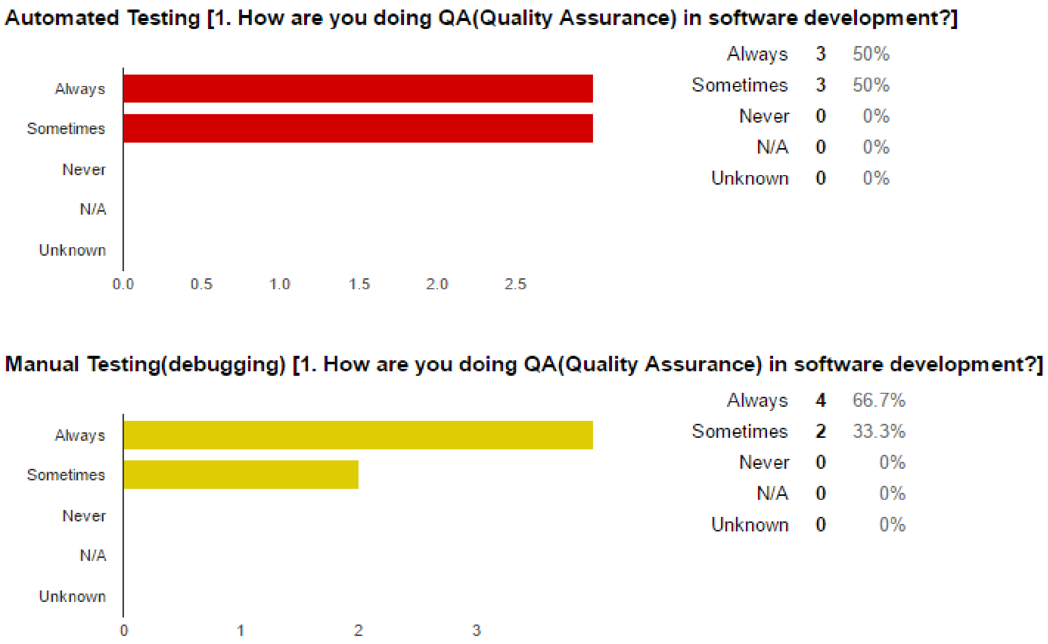
**Figure A.8:** Response analysis of the questions "17. What type of company are you currently in?" and "18. What is your job role?"

- **IT/Software Companies(Staff group):**



| | | |
|---|---|---|
| Typical project | 1 | 16.7% |
| Typical product | 2 | 33.3% |
| Purchase/evaluation | 0 | 0% |
| Typical to company | 3 | 50% |
| Other | 0 | 0% |

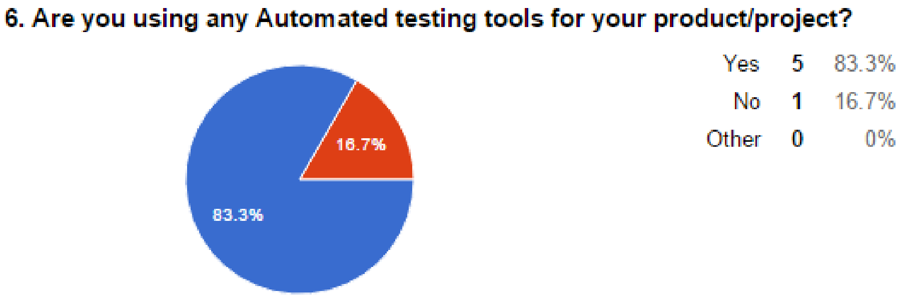
**Figure A.9:** Response analysis of the question “Select scope of answer”



| | | |
|---|---|---|
| Always | 3 | 50% |
| Sometimes | 3 | 50% |
| Never | 0 | 0% |
| N/A | 0 | 0% |
| Unknown | 0 | 0% |

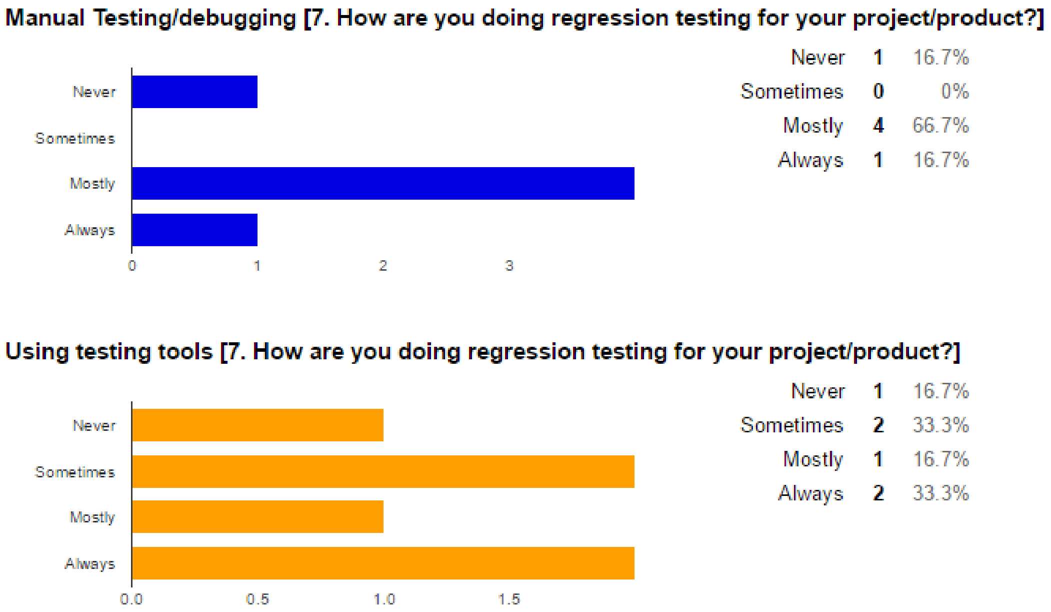
| | | |
|---|---|---|
| Always | 4 | 66.7% |
| Sometimes | 2 | 33.3% |
| Never | 0 | 0% |
| N/A | 0 | 0% |
| Unknown | 0 | 0% |

**Figure A.10:** Response analysis of the question “1. How are you doing QA(Quality Assurance) in software development?”



| | | |
|---|---|---|
| Yes | 5 | 83.3% |
| No | 1 | 16.7% |
| Other | 0 | 0% |

**Figure A.11:** Response analysis of the question “6. Are you using any Automated testing tools for your product/project?”



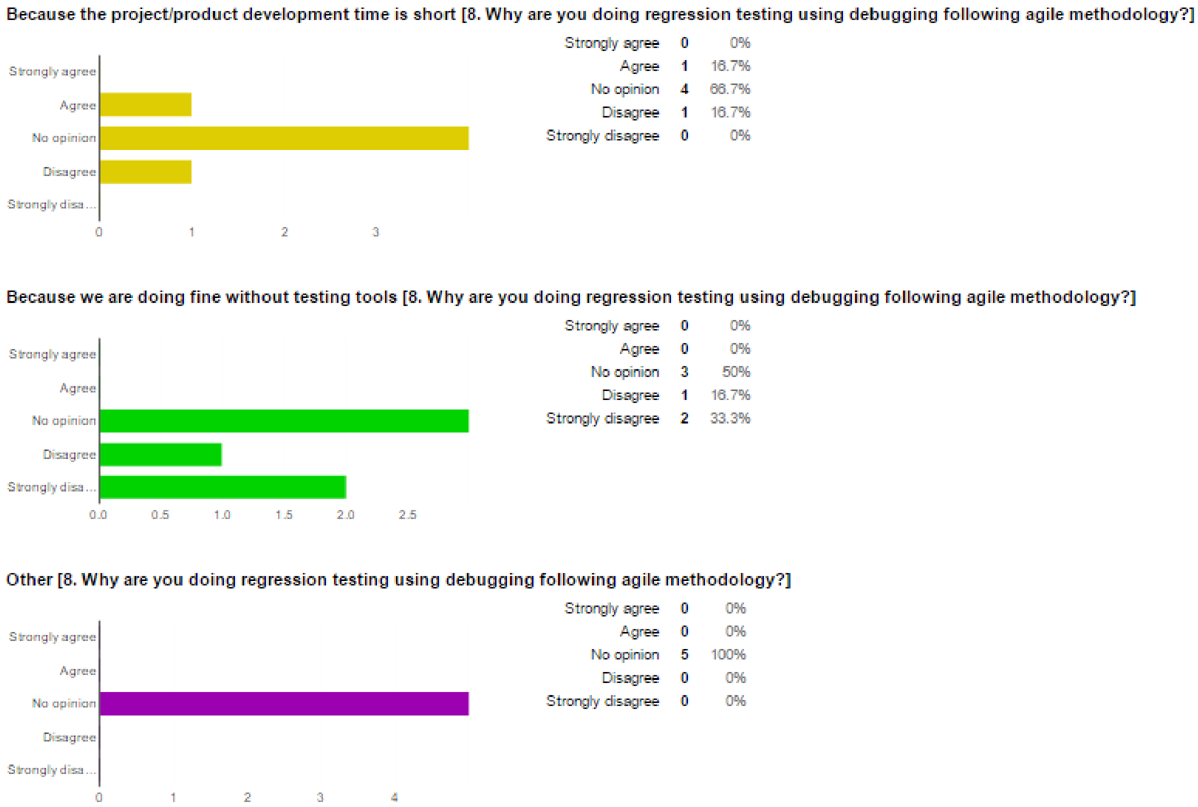
Manual Testing/debugging [7. How are you doing regression testing for your project/product?]

| | | |
|---|---|---|
| Never | 1 | 16.7% |
| Sometimes | 0 | 0% |
| Mostly | 4 | 66.7% |
| Always | 1 | 16.7% |

Using testing tools [7. How are you doing regression testing for your project/product?]

| | | |
|---|---|---|
| Never | 1 | 16.7% |
| Sometimes | 2 | 33.3% |
| Mostly | 1 | 16.7% |
| Always | 2 | 33.3% |

**Figure A.12:** Response analysis of the question “7. How are you doing regression testing for your project/product?”



Because the project/product development time is short [8. Why are you doing regression testing using debugging following agile methodology?]

| | | |
|---|---|---|
| Strongly agree | 0 | 0% |
| Agree | 1 | 16.7% |
| No opinion | 4 | 66.7% |
| Disagree | 1 | 16.7% |
| Strongly disagree | 0 | 0% |

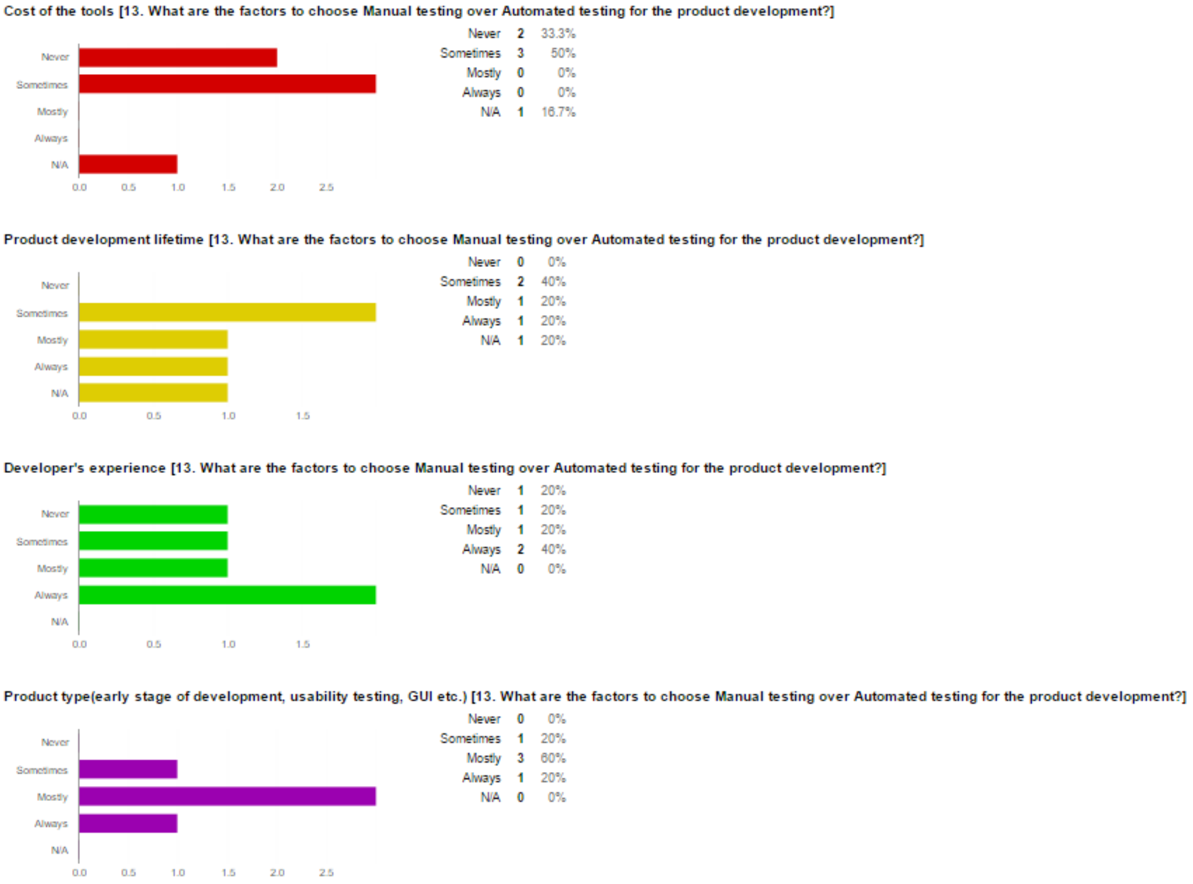
Because we are doing fine without testing tools [8. Why are you doing regression testing using debugging following agile methodology?]

| | | |
|---|---|---|
| Strongly agree | 0 | 0% |
| Agree | 0 | 0% |
| No opinion | 3 | 50% |
| Disagree | 1 | 16.7% |
| Strongly disagree | 2 | 33.3% |

Other [8. Why are you doing regression testing using debugging following agile methodology?]

| | | |
|---|---|---|
| Strongly agree | 0 | 0% |
| Agree | 0 | 0% |
| No opinion | 5 | 100% |
| Disagree | 0 | 0% |
| Strongly disagree | 0 | 0% |

**Figure A.13:** Response analysis of the question “8. Why are you doing regression testing using debugging following agile methodology?”


Cost of the tools [13. What are the factors to choose Manual testing over Automated testing for the product development?]

| | | |
|---|---|---|
| Never | 2 | 33.3% |
| Sometimes | 3 | 50% |
| Mostly | 0 | 0% |
| Always | 0 | 0% |
| N/A | 1 | 16.7% |

Product development lifetime [13. What are the factors to choose Manual testing over Automated testing for the product development?]

| | | |
|---|---|---|
| Never | 0 | 0% |
| Sometimes | 2 | 40% |
| Mostly | 1 | 20% |
| Always | 1 | 20% |
| N/A | 1 | 20% |

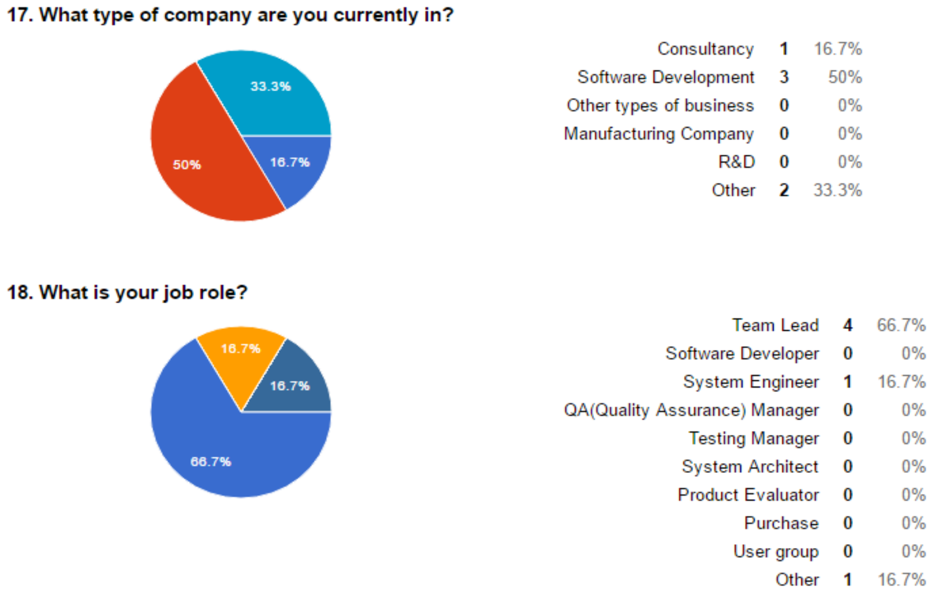
Developer's experience [13. What are the factors to choose Manual testing over Automated testing for the product development?]

| | | |
|---|---|---|
| Never | 1 | 20% |
| Sometimes | 1 | 20% |
| Mostly | 1 | 20% |
| Always | 2 | 40% |
| N/A | 0 | 0% |

Product type(early stage of development, usability testing, GUI etc.) [13. What are the factors to choose Manual testing over Automated testing for the product development?]

| | | |
|---|---|---|
| Never | 0 | 0% |
| Sometimes | 1 | 20% |
| Mostly | 3 | 60% |
| Always | 1 | 20% |
| N/A | 0 | 0% |

**Figure A.14:** Response analysis of the question "13. What are the factors to choose Manual testing over Automated testing for the product development?"


17. What type of company are you currently in?

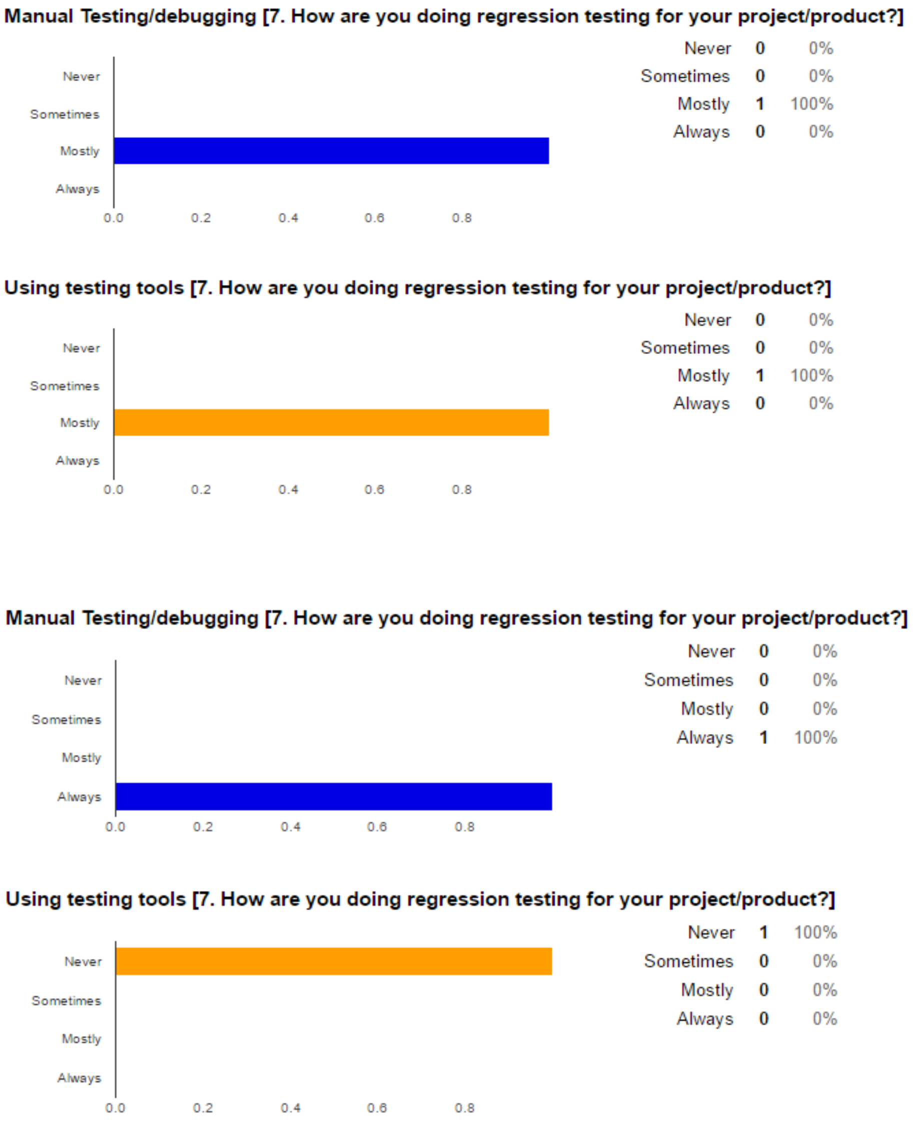
| | | |
|---|---|---|
| Consultancy | 1 | 16.7% |
| Software Development | 3 | 50% |
| Other types of business | 0 | 0% |
| Manufacturing Company | 0 | 0% |
| R&D | 0 | 0% |
| Other | 2 | 33.3% |

18. What is your job role?

| | | |
|---|---|---|
| Team Lead | 4 | 66.7% |
| Software Developer | 0 | 0% |
| System Engineer | 1 | 16.7% |
| QA(Quality Assurance) Manager | 0 | 0% |
| Testing Manager | 0 | 0% |
| System Architect | 0 | 0% |
| Product Evaluator | 0 | 0% |
| Purchase | 0 | 0% |
| User group | 0 | 0% |
| Other | 1 | 16.7% |

**Figure A.15:** Response analysis of the questions "17. What type of company are you currently in?" and "18. What is your job role?"

- **Non-IT/Software Companies(CEO & Staff groups):**



Manual Testing/debugging [7. How are you doing regression testing for your project/product?]

| | | |
|---|---|---|
| Never | 0 | 0% |
| Sometimes | 0 | 0% |
| Mostly | 1 | 100% |
| Always | 0 | 0% |

Using testing tools [7. How are you doing regression testing for your project/product?]

| | | |
|---|---|---|
| Never | 0 | 0% |
| Sometimes | 0 | 0% |
| Mostly | 1 | 100% |
| Always | 0 | 0% |

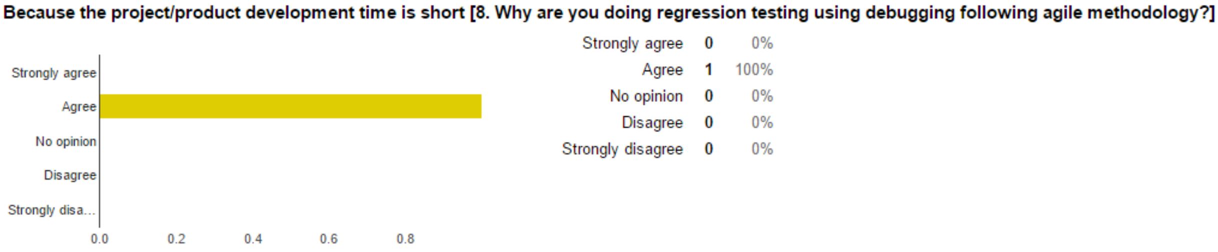
Manual Testing/debugging [7. How are you doing regression testing for your project/product?]

| | | |
|---|---|---|
| Never | 0 | 0% |
| Sometimes | 0 | 0% |
| Mostly | 0 | 0% |
| Always | 1 | 100% |

Using testing tools [7. How are you doing regression testing for your project/product?]

| | | |
|---|---|---|
| Never | 1 | 100% |
| Sometimes | 0 | 0% |
| Mostly | 0 | 0% |
| Always | 0 | 0% |

**Figure A.16:** Response analysis of the question “7. How are you doing regression testing for your project/product?” from both CEO and employee group of non-IT/software companies



Because the project/product development time is short [8. Why are you doing regression testing using debugging following agile methodology?]

| | | |
|---|---|---|
| Strongly agree | 0 | 0% |
| Agree | 1 | 100% |
| No opinion | 0 | 0% |
| Disagree | 0 | 0% |
| Strongly disagree | 0 | 0% |

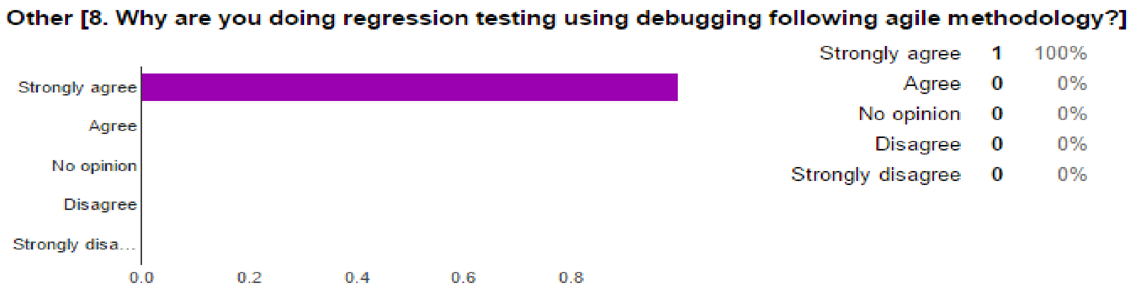
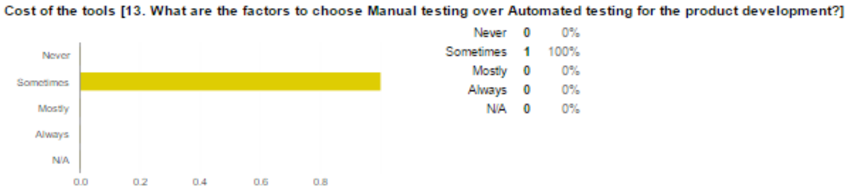
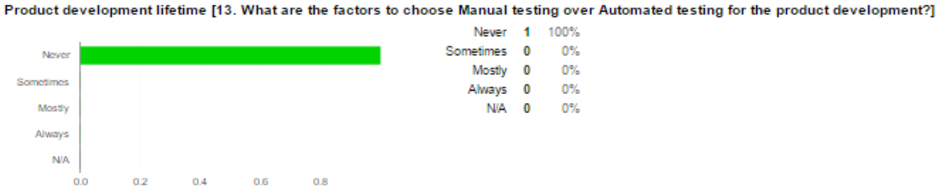
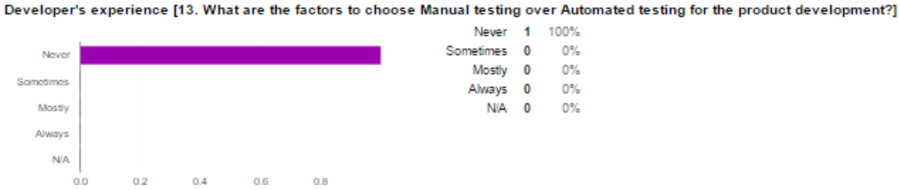
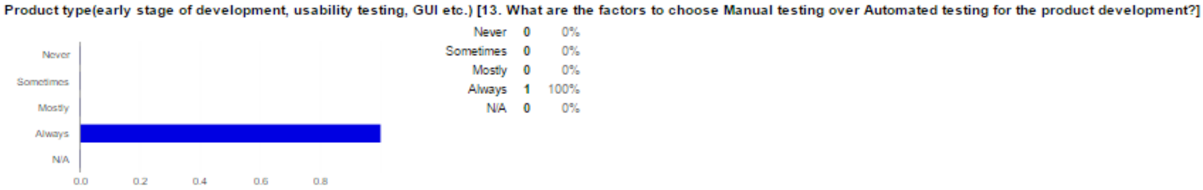


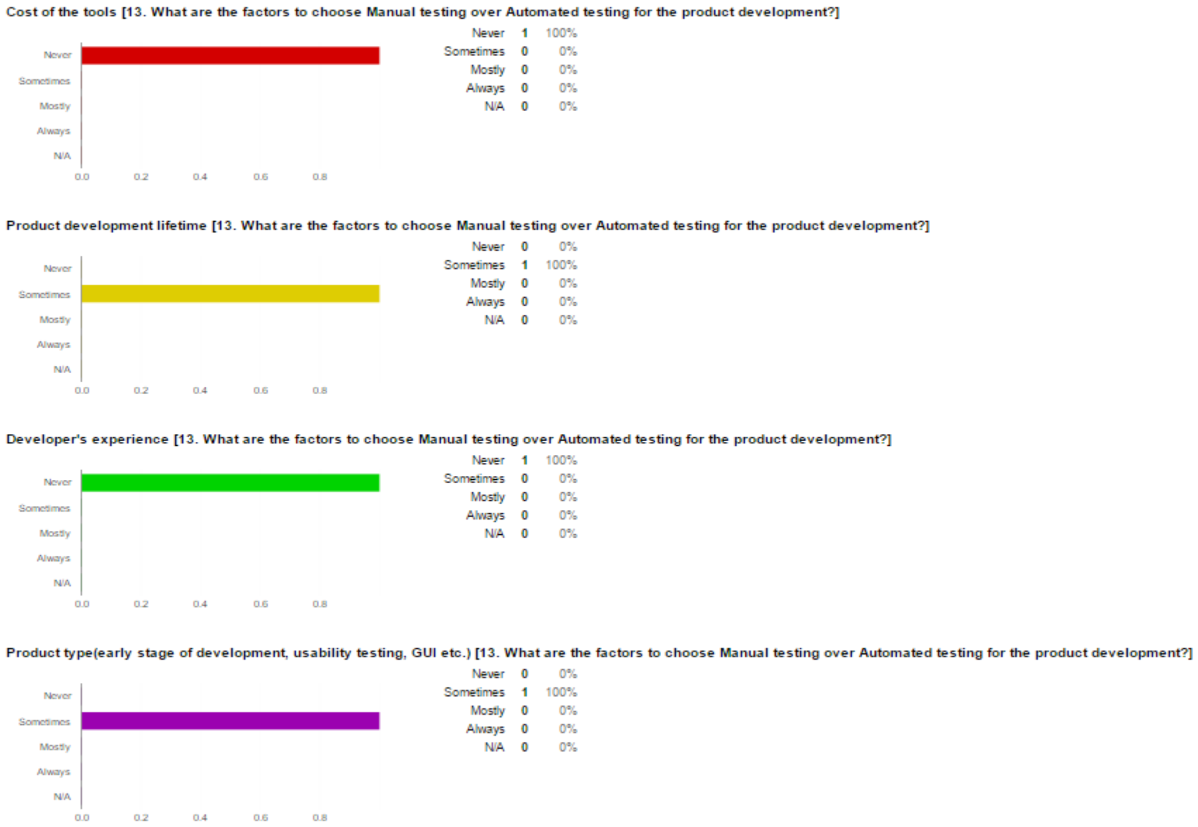
**Figure A.17:** Response analysis of the question "8. Why are you doing regression testing using debugging following agile methodology?" from both CEO and employee group of non-IT/software companies











Cost of the tools [13. What are the factors to choose Manual testing over Automated testing for the product development?]

| | | |
|---|---|---|
| Never | 1 | 100% |
| Sometimes | 0 | 0% |
| Mostly | 0 | 0% |
| Always | 0 | 0% |
| N/A | 0 | 0% |

Product development lifetime [13. What are the factors to choose Manual testing over Automated testing for the product development?]

| | | |
|---|---|---|
| Never | 0 | 0% |
| Sometimes | 1 | 100% |
| Mostly | 0 | 0% |
| Always | 0 | 0% |
| N/A | 0 | 0% |

Developer's experience [13. What are the factors to choose Manual testing over Automated testing for the product development?]

| | | |
|---|---|---|
| Never | 1 | 100% |
| Sometimes | 0 | 0% |
| Mostly | 0 | 0% |
| Always | 0 | 0% |
| N/A | 0 | 0% |

Product type(early stage of development, usability testing, GUI etc.) [13. What are the factors to choose Manual testing over Automated testing for the product development?]

| | | |
|---|---|---|
| Never | 0 | 0% |
| Sometimes | 1 | 100% |
| Mostly | 0 | 0% |
| Always | 0 | 0% |
| N/A | 0 | 0% |

**Figure A.18:** Response analysis of the question "13. What are the factors to choose Manual testing over Automated testing for the product development?" from both CEO and employee group of non-IT/software companies

## Appendix B: The list of questions for the interview

1. When do you start QA in your projects? If you can elaborate on the process
2. How popular is the use of testing tools in your company?
    a. (if popular) Why do you think it is the way it is? What are the factors behind it?
        I. Which tools do you use usually for your projects/products?
    b. (if not popular) Why do you think it is not? What are the factors behind it?
        II. Do you think this present scenario will change gradually in the near future?
3. Which kind of development process are you using?
    a. How do you do testing?
    b. What is your perspective on doing manual testing following agile methodology?
4. How are you doing regression testing for your projects/products? Which tools you use for it?
    a. (If not using tools) Do you follow agile methodology while doing regression testing manually? If you can elaborate on the process
5. How often do you do code coverage testing? Which tools do you use for it? How much satisfied are you with the tool's performance?
    a. (if not using tools) How do you do it manually? How much satisfied are you with the testing results?
    b. How important is it for your projects/products?
    c. What is your perspective on input validation testing, domain testing?
6. What are the factors your company consider to choose tools for project/product?
    a. How much satisfying performance do they give?
    b. What are reasons that influence your testing efforts?
    c. In which situations(GUI testing, usability testing etc.) manual testing overpowers testing tools? If you can elaborate on this
7. What is the most challenging situation you had during testing?
    a. What do you do when there is not enough time for thorough testing?

## Appendix C: Questionnaire for the participants of the survey

The questionnaire sample that has been used for the investigation of the use of test automation in software quality assurance in Norwegian companies and organisations is given below:

Edit this form

# Investigation of the use of Test Automation in Software Quality Assurance in Norwegian Companies and Organisations

This survey is a part of my Masters thesis which is on the practice of Automated Testing in Norwegian companies under the supervision of Senior Lecturer Carl-Fredrik Sørensen from Department of Computer and Information Science, Norwegian University of Science and Technology(NTNU). It focuses on the use of testing tools in the development of software products in different companies. Testing is inevitable for getting a bug-free end product. Automated and manual testing both are quite popular and some companies use both, some use mostly manual testing. This questionnaire is divided in 7 sections each containing a number of questions to get feedback of the use of automated software testing in Norway. Please select a scope of your answers: project perspective, product perspective or other. It should take no longer than 10 minutes to finish answering this investigation. Thank you in advance for taking your time to participate in this survey. All the answers along with your personal information will be treated confidentially.

* Required

**Select scope of answer ***

It is to determine whether your answers are from project perspective, product perspective or others

○ Typical project

○ Typical product

○ Purchase/evaluation

○ Typical to company

○ Other: ________

## Testing Method

This section is about which testing methods are mainly used in the development of software products.

**1. How are you doing QA(Quality Assurance) in software development?**

| | Always | Sometimes | Never | N/A | Unknown |
|---|---|---|---|---|---|
| Reviews | ○ | ○ | ○ | ○ | ○ |
| Inspection | ○ | ○ | ○ | ○ | ○ |
| Automated Testing | ○ | ○ | ○ | ○ | ○ |
| Manual Testing(debugging) | ○ | ○ | ○ | ○ | ○ |

### 2. Which type of validation/verification strategy do you follow for your product/project?

| | Never | Sometimes | Mostly | Always |
|---|---|---|---|---|
| Formal Methods | ○ | ○ | ○ | ○ |
| Syntactic Methods(Used in systems that are not mission-critical) | ○ | ○ | ○ | ○ |
| Test Cases(prepared/standard) | ○ | ○ | ○ | ○ |
| Customer Complaints | ○ | ○ | ○ | ○ |
| Reviews | ○ | ○ | ○ | ○ |
| Quality Manual | ○ | ○ | ○ | ○ |
| Configuration Management/issue tracker/backlog | ○ | ○ | ○ | ○ |
| Others | ○ | ○ | ○ | ○ |

### 3. Which of these test types or methods are used in your testing processes?

| | Never | Sometimes | Mostly | Always |
|---|---|---|---|---|
| Regression Testing | ○ | ○ | ○ | ○ |
| Installation Testing | ○ | ○ | ○ | ○ |
| Compatibility Testing | ○ | ○ | ○ | ○ |
| Acceptance Testing | ○ | ○ | ○ | ○ |
| Alpha Testing | ○ | ○ | ○ | ○ |
| Beta Testing | ○ | ○ | ○ | ○ |
| Functional Testing | ○ | ○ | ○ | ○ |
| Non-functional Testing | ○ | ○ | ○ | ○ |
| Usability Testing | ○ | ○ | ○ | ○ |
| Security Testing | ○ | ○ | ○ | ○ |
| Static Testing | ○ | ○ | ○ | ○ |
| Dynamic Testing | ○ | ○ | ○ | ○ |
| White-box Testing | ○ | ○ | ○ | ○ |
| Grey-box Testing | ○ | ○ | ○ | ○ |
| Black-box Testing | ○ | ○ | ○ | ○ |

### 4. How do you trace test cases and test results to requirements and vice versa?

| | Always | Mostly | Sometimes | May be/Do not know | Never | N/A |
|---|---|---|---|---|---|---|
| Bi-directional Traceability Matrix(It maps requirements to test cases and vice versa) | ○ | ○ | ○ | ○ | ○ | ○ |
| Configuration Management System | ○ | ○ | ○ | ○ | ○ | ○ |
| Others(using tools like SmartExcel4TFS, TBreq etc.) | ○ | ○ | ○ | ○ | ○ | ○ |

## Development Process

This section is about which testing processes are used in the development of software products.

**5. Which development process is most used in the development of the software product in your company?**

You can choose multiple answers for this question.

| | Sometimes | Mostly | Always | Never |
|---|---|---|---|---|
| Agile Development Model(Scrum, Kanban, XP, Lean etc.) | ○ | ○ | ○ | ○ |
| Waterfall Development Model | ○ | ○ | ○ | ○ |
| DevOps | ○ | ○ | ○ | ○ |
| Spiral Development. Incremental Development etc. | ○ | ○ | ○ | ○ |
| Test Driven Development(XP) | ○ | ○ | ○ | ○ |
| Others | ○ | ○ | ○ | ○ |

## Tools

This section is about the tools that are being used in the testing of the software product.

**6. Are you using any Automated testing tools for your product/project?** *

○ Yes

○ No

○ Other: ______

**7. How are you doing regression testing for your project/product?**

| | Never | Sometimes | Mostly | Always |
|---|---|---|---|---|
| Manual Testing/debugging | ○ | ○ | ○ | ○ |
| Using testing tools | ○ | ○ | ○ | ○ |
| Other | ○ | ○ | ○ | ○ |

**8. Why are you doing regression testing using debugging following agile methodology?**

| | Strongly agree | Agree | No opinion | Disagree | Strongly disagree |
|---|---|---|---|---|---|
| Because the project/product development time is short | ○ | ○ | ○ | ○ | ○ |
| Because we are doing fine without testing tools | ○ | ○ | ○ | ○ | ○ |
| Other | ○ | ○ | ○ | ○ | ○ |

**9. Are you using any code coverage testing tool? Please mention the amount of coverage you are getting along with your experience with the tool. ***

**10. Which method do you consider to be best fit for testing**

| | Manual | Automated | Both | Never | N/A |
|---|---|---|---|---|---|
| Functionality | ○ | ○ | ○ | ○ | ○ |
| Security | ○ | ○ | ○ | ○ | ○ |
| Accuracy | ○ | ○ | ○ | ○ | ○ |
| Reliability | ○ | ○ | ○ | ○ | ○ |
| Recoverability | ○ | ○ | ○ | ○ | ○ |
| Usability | ○ | ○ | ○ | ○ | ○ |
| Learnability | ○ | ○ | ○ | ○ | ○ |
| Efficiency | ○ | ○ | ○ | ○ | ○ |

| | | | | | |
|---|---|---|---|---|---|
| Maintainability | ○ | ○ | ○ | ○ | ○ |
| Stability | ○ | ○ | ○ | ○ | ○ |
| Testability | ○ | ○ | ○ | ○ | ○ |
| Installability | ○ | ○ | ○ | ○ | ○ |
| Portability | ○ | ○ | ○ | ○ | ○ |
| Adaptability | ○ | ○ | ○ | ○ | ○ |
| Performance | ○ | ○ | ○ | ○ | ○ |
| Scalability | ○ | ○ | ○ | ○ | ○ |
| Interoperability | ○ | ○ | ○ | ○ | ○ |

# Experience

This section is about the experience your company have using the testing tool.

**11. How do you handle test bugs in the testing tool?**

| | Never | Sometimes | Mostly | Always | N/A |
|---|---|---|---|---|---|
| Drop the tool that was supposed to fix the bugs | ○ | ○ | ○ | ○ | ○ |
| Try to fix the bugs in the tool | ○ | ○ | ○ | ○ | ○ |
| Using bug tracking tools(Bugzilla, JIRA etc.) to handle the test bugs beforehand | ○ | ○ | ○ | ○ | ○ |
| Do not know | ○ | ○ | ○ | ○ | ○ |

**12. Which factors do you consider while selecting testing tools for test automation?**

| | Always | Mostly | Sometimes | Never | N/A |
|---|---|---|---|---|---|
| Cost | ○ | ○ | ○ | ○ | ○ |
| Popularity | ○ | ○ | ○ | ○ | ○ |
| Developer's experience | ○ | ○ | ○ | ○ | ○ |
| Functionality | ○ | ○ | ○ | ○ | ○ |
| Others | ○ | ○ | ○ | ○ | ○ |

## Manual Testing

Manual Testing is where you need to write test suites for your software application to perform the test on them manually instead of using testing tools.
If you are not using manual testing, then you do not have to answer the questions of this section.

**13. What are the factors to choose Manual testing over Automated testing for the product development?**

| | Never | Sometimes | Mostly | Always | N/A |
|---|---|---|---|---|---|
| Cost of the tools | ○ | ○ | ○ | ○ | ○ |
| Product development lifetime | ○ | ○ | ○ | ○ | ○ |
| Developer's experience | ○ | ○ | ○ | ○ | ○ |
| Product type(early stage of development, usability testing, GUI etc.) | ○ | ○ | ○ | ○ | ○ |
| Others(Ad-hoc testing, re-run of test cases etc.) | ○ | ○ | ○ | ○ | ○ |

**14. How often do you consider to re-use test suites for the testing purpose of your product?** *

| | 1 | 2 | 3 | 4 | 5 | |
|---|---|---|---|---|---|---|
| Never | ○ | ○ | ○ | ○ | ○ | Always |

## Documentation

This section refers to the documentation of testing processes and tools

**15. Do your company keep documentation of testing processes and tools you use in software development?**

| | Never | Sometimes | Mostly | Always | N/A |
|---|---|---|---|---|---|
| Test plan | ○ | ○ | ○ | ○ | ○ |
| Test design and Test case specification | ○ | ○ | ○ | ○ | ○ |
| Test Strategy | ○ | ○ | ○ | ○ | ○ |
| Test summary report | ○ | ○ | ○ | ○ | ○ |
| Weekly Status Report | ○ | ○ | ○ | ○ | ○ |
| User Documents/ manuals | ○ | ○ | ○ | ○ | ○ |
| User Acceptance | | | | | |

| | | | | | |
|---|---|---|---|---|---|
| Report | ○ | ○ | ○ | ○ | ○ |
| Risk Assessment | ○ | ○ | ○ | ○ | ○ |
| Test Log | ○ | ○ | ○ | ○ | ○ |
| Bug report | ○ | ○ | ○ | ○ | ○ |
| Test data | ○ | ○ | ○ | ○ | ○ |
| Test analysis | ○ | ○ | ○ | ○ | ○ |

**16. How do you manage test reports from automated testing?** *

○ Part of deployment documentation
○ Are not collected
○ Just for internal use
○ Other: ______

## Demographic Questions

**17. What type of company are you currently in?** *

○ Consultancy
○ Software Development
○ Other types of business
○ Manufacturing Company
○ R&D
○ Other: ______

**18. What is your job role?** *

○ Team Lead
○ Software Developer
○ System Engineer
○ QA(Quality Assurance) Manager
○ Testing Manager
○ System Architect
○ Product Evaluator
○ Purchase
○ User group
○ Other: ______

**19. Which department do you work in?** *

○ QA(Quality Assurance)
○ Development

○ Testing

○ Administrative

○ IT Advisory

○ Staff

○ Purchase

○ Other: __________

**20. How much of total population of your company are working in IT related sectors?**

| | 1-10 | 10-40 | 50-80 | >100 |
|---|---|---|---|---|
| Developers(in percentage) | ○ | ○ | ○ | ○ |
| Testers(in percentage) | ○ | ○ | ○ | ○ |
| Employees(in percentage) | ○ | ○ | ○ | ○ |

**21. Could you consider in doing an interview with us?**

Please leave your email address in the textbox and we will contact you

__________

Submit

*Never submit passwords through Google Forms.*

100%: You made it.

Powered by

This content is neither created nor endorsed by Google.

Report Abuse - Terms of Service - Additional Terms