



# **Bachelor Diploma**

## **Thesis**

**Faculty of economics and management**

### **Software testing**

**Supervisor: - Vojtech Merunka**

**Author: - Ankit Vaniya**

## BACHELOR THESIS ASSIGNMENT

Ankit Vaniya

Informatics

Thesis title

**Software testing**

---

### Objectives of thesis

The goal of thesis is to describe the most popular techniques used during stages of various methodologies of the software development lifecycle. The second part of thesis will be practical example of a test design techniques.

### Methodology

Use common software engineering standards such as use-case of the UML. Describe testing levels (unit, system, integration, user-acceptance testing). Do not forget to write an example of test documentation. Thesis will cover all three main software platforms: Windows, Android, iOS, OSX, UNIX-Linux.

## **The proposed extent of the thesis**

30-40 pages

## **Keywords**

Software testing, development life cycle, quality control, testing fundamentals

---

## **Recommended information sources**

Brian Marick, *The Craft of Software Testing*, Prentice Hall, 1995.

Cem Kaner, "Software Negligence & Testing Coverage," in *Proceedings of STAR 96, (Software Quality Engineering, Jacksonville, FL)*, 1996.

C. Kaner, J. Falk, and H.Q. Nguyen, *Testing Computer Software (2/e)*, Van Nostrand Reinhold, 1993.

M. Friedman and J. Voas, *Software Assessment: Reliability, Safety, Testability*, Wiley, 1995.

Michael Dyer, *The Cleanroom Approach to Quality Software Development*, Wiley, 1992.

---

## **Expected date of thesis defence**

2017/18 SS – FEM

## **The Bachelor Thesis Supervisor**

doc. Ing. Vojtěch Merunka, Ph.D.

## **Supervising department**

Department of Information Engineering

Electronic approval: 23. 2. 2018

---

**Ing. Martin Pelikán, Ph.D.**

Head of department

Electronic approval: 23. 2. 2018

---

**Ing. Martin Pelikán, Ph.D.**

Dean

Prague on 15. 03. 2018

---

## **Declaration**

I declare that I have worked on my diploma thesis titled “Software Testing” by myself and I have used only the sources mentioned at the end of the thesis. As the author of the diploma thesis, I declare that the thesis does not break copyrights of any their person.

In Prague on **15 March 2018** \_\_\_\_\_

## **Acknowledgement**

I declare that the thesis “Software testing” is developed independently under the guidance of my supervisor, Doc.Ing Vojtech Merunka, PH.D for which I am thankful to him and other person for their advice and support given during my research. I also declare that the use of literature and other information sources that are cited in the work are listed in the reference at the end of work. As the author of the referred thesis, I further declare that I didn't infringe the copyright of any third parties in the context of its creation.

## Shrnutí

Tato práce se zabývá informacemi o testování softwaru jako studie o životním cyklu vývoje softwaru a jak funguje proces vývoje softwaru a jaké jsou fáze a jak u nich postupovat. Dále tato práce poskytuje informace o krocích testování, technikách návrhu testů, typech testů, procesech odhadu testů, testovací dokumentaci, testovacích strategiích, testovacích nástrojích, metodikách vývoje softwaru, nasazení softwaru, chybách v životním cyklu, automatizaci testů apod.

Proces testování softwaru je jednou z nejdůležitějších fází procesu vývoje softwaru, kam se aplikuje podle modelu životního cyklu vývoje softwaru, kterými jsou vodopádový model, V-model, iterativní, spirální model a agilní modely. Každý model má svoji vlastní metodiku pro vývoj softwaru. Proces testování je sice vždy stejný, ale postavení fáze testování se v jednotlivých metodikách liší. Testování softwaru je proces jeho ověřování a validace. Testování softwaru se řídí dvěma typy jako manuální testování a nebo automatizované testování. Manuální testování provádí ručně tester. Automatizované testování vyžaduje potřebné testovací nástroje.

**Klíčová slova:** Životní cyklus vývoje softwaru, metodologie vývoje softwaru, modely vývoje softwaru, typy testů, techniky návrhu testů, hodnocení testů, úroveň testů, strategie testů, testovací nástroje, testovací dokumentace, chyby v životním cyklu, atd.

## Summary

This thesis is conducted the information about the software testing. The study about software development life cycle. How software development process works and what stages are there and how to follow each and everything is conducted by this thesis. Moreover this thesis provide an information about testing step, test design technique, testing types, test estimation process, testing documents, test strategy, testing tools, software development methodology, software deployment, bug life cycle, test automation etc.

The software testing process is one of the most important phase of software development process. In this phase is apply as per model of software development life cycle. There are such like water fall, v model, iterative, spiral and agile. Every model have different type of methodology to develop software. The testing process is the same but phase of testing is different in all software development models. Software testing is process of verification and validation. Software testing follow two types as manual testing and automation testing. Manual testing is done by manually by the tester. Automation testing needs testing tools.

**Keywords:** Software development life cycle, software development methodology, software development models, test type, test designs technique, test estimation, test levels, test strategy, test tools, test documentation, bug life cycle etc.

# Table of Content

1) Introduction.....	1
1.1. What is software testing.....	1
1.2. Fundamental of testing.....	1
1.3. Software testing principle.....	5
1.4. Important of testing.....	6
1.5. Who does testing.....	7
1.6. When to start testing.....	8
1.7. When to finish testing.....	8
2) Goal of thesis.....	9
3) Methodology of thesis.....	10
4) Software development life cycle.....	10
4.1. Software development model.....	11
4.1.1. Water fall model.....	11
4.1.2. Spiral model.....	16
4.1.3. V-model.....	19
4.1.4. Iterative model.....	23
5) Software testing levels.....	25
5.1. Unite testing.....	25
5.2. Integration testing.....	25
5.3. System testing.....	26
5.4. Acceptance testing.....	27
6) Software test design technique.....	28
6.1. Static test design technique.....	28
6.1.1. Manual static test design technique.....	28
6.1.2. Static test design technique using tools.....	29
6.2. Dynamic test design technique.....	30
6.2.1. Specification based test design technique.....	30
6.2.2. Structured based test design technique.....	31
6.2.3. Experience based test design technique.....	32
7) Testing types.....	33
8) Test estimation methods.....	40
8.1. Work break down test estimation method.....	40
8.2. Three point test estimation method.....	41
8.3. Planning poker test estimation method.....	42
9) Test strategy.....	43
10) Test documentation.....	45
10.1. Test plan.....	45
10.2. Test case.....	46
10.3. Bug report.....	48
10.4. Test summary report.....	48



11)	Bug life cycle.....	50
12)	Automation testing.....	52
12.1.	Automation testing tools.....	52
12.1.1.	Testing tools for functional testing.....	52
12.1.2.	Testing tools for non-functional testing.....	52
13)	Appendix.....	53
14)	Conclusion.....	66
15)	References.....	67

## List of Figures

Figure 1: Who does testing.....	8
Figure 2: Waterfall model .....	12
Figure 3: Spiral Model.....	17
Figure 4: V Model.....	21
Figure 5: Iterative Model.....	23
Figure 6: Graphical user interface testing.....	37
Figure 7: Test estimation method.....	40
Figure 8: Work breakdown estimation method.....	41
Figure 9: Test strategy documents.....	43
Figure 10: Bug life cycle.....	50

# Introduction

## What is software testing?

The software testing has various definition and views. Testing process is the process of evaluating the software or its component and functionalities with the intention to find whether it specified as per the requirement or not. In other words we can say that testing is validation and verification process of the software. There are few more concepts to define as testing is executing a software or program in order to identify bugs or missing requirement that is different to the actual requirement or additionally we can say that testing is process that need to check system or software is behave or not as it should be. Moreover a process of analysis a program or software to find difference between existing and required condition and to check the features of program or software. (Michael Dyer, 1992)

Following authors cited above, we can say that Software testing process has main two elements as per below:-

- **Validation**

Validation is process is to check that program or software will meets and satisfy the specific requirement. In other words software or program should develop as per customer requirement.

- **Verification**

Verification process is to check that the program or software satisfies the requirements imposed as the start of the development part. In other words the program or software should behave as we want it to.

## Fundamental of testing

Software testing is rather than single process. The software testing process starts from gathering information of requirements then need to test first requirements that the given requirement is fully correct and enough. Requirement should have fullness, unique, testable, relational and accessible now it is time to panning of software as per we got requirements. After that need to analyze and design that as

per planning. Now need to check for implementation in case we can implement on something after this there will be on next stage is execution. Execution and evaluating are the final stage of fundamental of testing. So we can divide the fundamental of testing in to following basic stapes as per below.(Müller, Friedenber, Verma, & Veenendaal, 2007)

## **Planning and control**

The test planning and control is the main first phase of software testing. In this phase needs to determine the scope and risk of the testing then identify the testing objective. Next needs to determine the test approaches. After this done needs to plan test strategy.

Test strategy is created to inform project manager, testers and developers about the some key issues of testing process. Test strategy include that objective that need to tested, method of testing, time limit and resources need to testing. The resources are like staff, test environment, test tools and computes. Also in this phase we need to set criteria such as coverage criteria. Coverage criteria will help us track whether we completing the test activities correctly and also show us that which task and check we need to complete for particular stage before we can say that testing is finished.

The test control also includes tasks. We need to analyze result of review and testing. We also need provide information on testing. Moreover we need to initiate correct actions and need to makes decisions. In last we need monitor and documents progress, test coverage and test exit criteria. (Müller, Friedenber, Verma, & Veenendaal, 2007)

## **Analysis and design**

Analysis and design is second phase. It is also include major tasks. In this phase we need to review the test basis. The test basis includes the information that we need to start test analysis and create the test cases. Basically this document is based on test cases. Test case is based on requirement, design specification, risk analysis, interface and architectures. Test base describe how system should behave once it build.

In Analysis and design phase we need to identify test conditions. Also we need to design the test and then need to evaluate testability of the requirement and system. Moreover we need to design test environment set-up and identify and required tools.(Müller, Friedenber, Verma, & Veenendaal, 2007)

## **Implementation and execution**

During the implementation and execution we need to consider test condition into test cases, procedures and other test ware like test scripts, test environment and test infrastructure. Test care is set of condition where tester will follow test and check whether it's going right or wrong. Test ware is term for all facilities that provided together for testing a program or software like test scripts, environment and infrastructure for later reuse. (Müller, Friedenber, Verma, & Veenendaal, 2007)

### **Test implementation has following task:**

- 1) To develop and priorities test cases by using test design technique and create test data. For test any application or software you need to input the data. This data should be specified of the feature which you are going to test. This data called test data. Sometime we need to test some application by automation so we need to input some test scripts for automation testing.
- 2) To create test suites from the test case for efficient execution. Test suites is the set of test case use to test software or program to show that it have same and specified behavior. Test suites also have detailed information on for each set of test suites on the system configuration to use while testing. Test suites are used to group same test together.
- 3) To implement and verify the environment.

### **Test execution has following task**

- 1) To execute test suites and individual test case
- 2) To re-execute the test case that failed to in order to verify a fix. This call confirmation testing or retesting.
- 3) To log the outcome of test execution and record the version of software which was under the test. The test log is used for audit trial. Test log is

nothing but the information about who executed test, which test case are executed, in what order we executed and what is result of executed test cases like pass or fail. This detailed information documentation called test log.

- 4) To compare actual results with expected results.
- 5) To verify that with test case passed and failed means difference between actual and expected result it should be report as bug.

## **Evaluating exit criteria and reporting**

- Evaluation exit is depends on risk assessment of the project. We need to set the criteria for each testing level against and need to measure “enough testing”. These criteria vary from project to project and are known as exit criteria. Exit criteria come when maximum test cases are executed with pass percentage, bugs are fall below certain rates and project is near to deadlines. (Michael Dyer, The Cleanroom Approach to Quality Software Development, Wiley, 1992.)

### **Evolution exit criteria tasks**

- 1) To check test logs and exit criteria specified in test planning.
- 2) To check in case more tests are needed or incase exit criteria should be changed.
- 3) To write test summery report.

## **Test closure activities**

Test closing activities are done when software is going to deliver to customer. The testing can also closed for other reason too like,

- All information we have which needed for testing.
- Incase project cancelled.
- Incase target achieved.
- Incase maintenance release or update is done.

## **Test closure activities tasks**

- To verify that which plans are already delivered and all incident reports are resolved.
- To verify test ware like test scripts, environment, etc for reuse.
- To handover test ware to maintenance for support and solution.
- To evaluate how was testing and what lessons learned.

## **Software testing principle**

### **1. Show defect in product.**

We are doing testing to find out the defects development. Testing cannot able to prove that product is defect free even they failed to find out the defects. We do testing to fix and correct the defect and make our product as per user requirements but testing is not proof of perfectness. Testing is the only process to find out the bugs even testing is done that not mean our product is 100% pure or bug free.

### **2. Exhaustive testing is not possible.**

Testing process is combination of all kind of inputs, precondition and preconditions. In exhaustive testing we need to test combination of all this kind inputs as per product requirement. For example we need to test a box where we can input only number. According to requirement it's should be take number between 1 to 100. It means it should not be take any number as input more than 100 and less than 1. Button should be work as per requirement. We do not have time to test for number between 1 to 100. That why we need to set criteria as we can check only boundary value so we do not need to each and every numbers.

### **3. Prior testing**

The software development life cycle have main part of testing.

Testing shall be starts from the beginning of requirement gathering and as soon as possible. Prior testing makes good impact of whole process. It can save us from waste of money and time. Prior testing should be focused more on requirements and objectives.

#### **4. Errors clustering**

Testing process should be focused on module wise. There should be lots of defect found because of small mistakes. For example in case there is 30% of code is wrong but because of that its effects very bad and too much on that left 70% of work. If we know that where we have made mistakes we do not need to fix all defects and we can find out more defects at similar module, class or area of coding.

#### **5. Repeat testing**

Repeat testing is most important part of testing. Some time we need to test again and again for the same thing but when we fix bug some time the test case will not be stay longer same as it was. To overcome this repeat testing always reviewed and test again and again for same set of test case also for new test in order to find out more defect and maintain quality of product.

#### **6. Non-appearance of defect is not true.**

If someone say that software is 100% correct that is not true. There is always some defects left but it's not come in our eyes because it is not make any impact on outputs. Testing should be more focus only on our product should be behaved as it should and it should be built as per customer requirements.

#### **7. Testing is always depends on requirements**

Testing process should be focus on what we are going to test. Testing should be done in differently for different types of testing. For example performance testing and functional testing should be tested on different criteria. Testing process always depends on requirement of testing types. (Software Testing by Gerald D. Everett and Raymond McLeod Jr.)

### **Importance of testing:**

The software testing is plays vital role in software development life cycle process. There was information mention about survey of software errors cost US economy

0.6 % of gross domestic product and about 80 % of software development costs of project are spent on finding and correction of defects. Usually in testing process product passed through some time more or same phases. So people prefer to early testing to avoid big problem. In test process because of early testing we can know about risk and to face it and prepare for it.

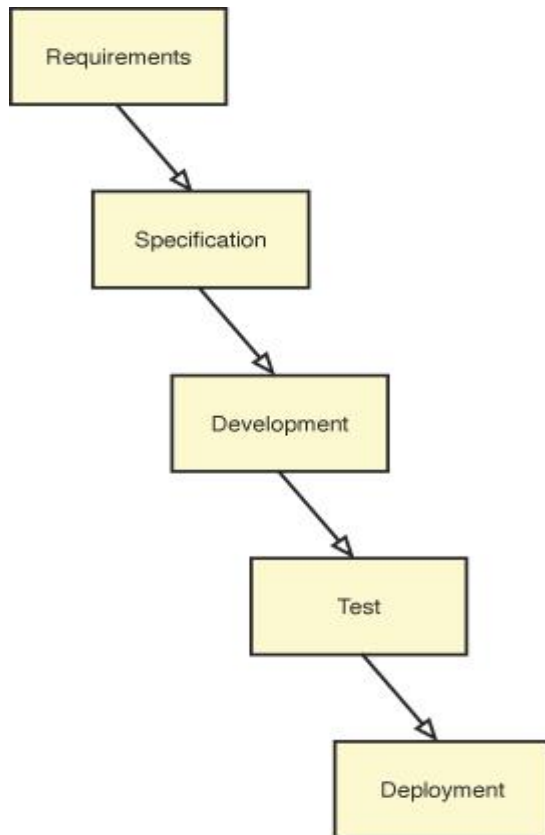
There are different phases in software development life cycle as per below:

1. **Test analysis:** to understand about requirement and need of customer.
2. **Test design:** create test case, scenario, scripts and environment as per customer requirement.
3. **Test execution:** test the test case and reviewed it incases if there are any errors.

### **Who does testing?**

The software testing is depends on the methodology that who are going to use in development of your product. In early days software engineers were use the waterfall model to develop the software. As per waterfall model first phase is to gathering requirements then study or specification on that then start to develop according that given requirements. After this whole process before you hand over this product to the customer it is going to hand over to testers to test and check the quality of software. After that in case there is any defect found it returns to developers to fix it than again come back to re testing. After it working well software is going for deployment. So software tester is most depended person who responsible for testing process but also sometime developer, project lead or manager and end user can do testing. You can see and understand as per figure.





### **When to start?**

Testing should start early as early as can start because it could save more time and cost. In software development life cycle testing should be start form the business requirement gathering and should be finished before the deployment phase. Testing process is always depends on the software development model being in used.

### **When to finish?**

If someone say that software is 100% pure it is not possible. There is always some bugs are there but it not effect on system and it is ignore by the testers. Testing should be finished ones all bugs with high priority is fixed. Also testing can be finished once rate of bugs are too small and duration of project is going to finish. Moreover also we can stop testing in case risk is low level than the limit. Testing is never ending process.

## **Goal of thesis**

The software testing thesis's goal is to understand the software development life cycle. This thesis provide information about to develop, test and deployment and quality control of intended software. There are all information about the manual testing. Also in this thesis can help to follow necessary criteria for testing a software as test design technique, test step, test type and test estimation as well as to prepare documentation on testing like test case, test report, bug report and test plan. In the thesis will be included a practical example of test design techniques.

## **Methodology of thesis**

This software testing thesis include the methodology of software development process including all necessary software engineering standards. Which are such as water fall model, spiral model, v-model and iterative model. Moreover this thesis includes information about software test design technique. Also about test step which testing process follow such as unite test, integration test, system test and acceptance test. This thesis provide information about test estimation methods like work breakdown, three point and planning poker. Furthermore this thesis have sample and templates of test document which are test case, test summery reports, bug repot and test plan on examples of all main operating system platforms.

## **Software development life cycle**

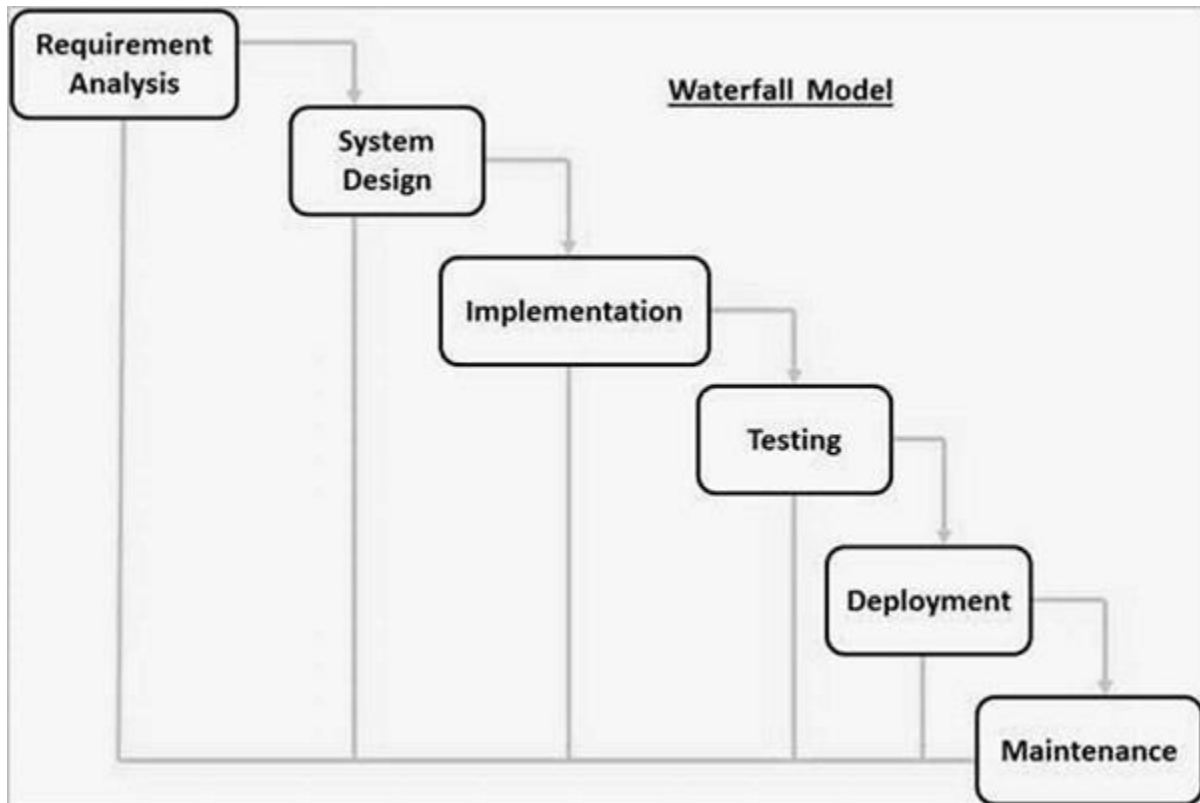
Software development life cycle is well designed, structured sequence of stages in software engineering to develop intended software. As per software development life cycle first stage is communication between the client and development team. Then after communication they decided to gather requirement as per client need. After they study and analyze requirement and check that it is relational with our project. Next step is analysis of system. After that design of software as per gather requirement. Further step is coddng for development. After this need to test the developed software and integration of software. Next step is implementation of software and after then maintenance in case of need. Last stage is deployment of software. For all this stage in software development life cycle developed different model as per requirement and project. That software development models are as per below.

# Software Development Life Cycle Models

## 1) WATERFALL MODEL

Waterfall model is the first process model in SLDC (Software Development Life Cycle). It is also known as a Linear-sequential life cycle model. The waterfall model is earliest approach that is use for any software development. The waterfall model is nothing but a simple software development process in linear sequential flow. Waterfall model is always describe in linear sequential way. Main condition or we can say necessary thing in waterfall model is each phase must be completed before the next phase begin and there is no overlapping in the phases allow in waterfall model. Any phases in the development process begins only if the previous phase is complete. Also we can say that waterfall model is a process in which progress is referred as a flowing increasingly downwards though a list of phases that must be executed in order to successfully build any computer software.

Waterfall model is the first Software development life cycle model that is used in software engineering to ensure the success of the any software development project. At first, the Waterfall model is developed by the Winston W. Royce in 1970 to describe the possible software engineering practice. The waterfall model consists of several number of phases. The whole process of software development is divided in separate phases. In this moving to next phase is possible only when its preceding phase is completely done. Because of this we can say that waterfall model is recursive in that each phase can be endlessly repeated until it is perfected. In this the output of one phase act as input for the next phase sequentially. (Tutorialspoint, n.d.)



The figure depicts the different phases of the SLDC Waterfall model. The above figure shows the different phases of the Waterfall models. The waterfall model comprises main six phases. There are as Requirement analysis, System design, Implementation, Testing, Deployment, Maintenance

- a) **Requirement analysis:** It is also known as Software requirements specification. It is complete description of the behaviour of any software to be developed. All possible requirements of the system that need to be developed are captured in requirements analysis phase and it is documented in the requirement specific document. It helps system and business analysts to define various kinds of requirements like functional requirements or non-functional requirements. It includes properties like reliability, scalability, testability, availability, maintainability, and performance.

- b) **System design:** This is the second phase of the model. The requirement specification from the first phase and studied in this phase and the system design is prepared. We can also say that it is process of planning and problem solving for a software solution. System design phase helps developers or designers to define the plan for solution which includes various parts like algorithm design, software architecture design, database conceptual schema and data structure definition. This system design helps in defining the overall system architecture.
- c) **Implementation:** The requirement specifications from first phase are studied in this phase and the system design is prepared. Implementation phase refers for the realization of business requirements and it is also used for design specifications into a concrete executable program, database, website, or software component through programming and deployment. In implementation phase the real code is written and code is also compiled into an operational application in this phase. Database and text files are created in implementation phase. It is nothing but the process of converting the whole requirements and blueprints into a production environment. This system design helps in specifying various kinds of requirements like hardware requirements and system requirements. This is also useful in defining the overall architecture of any system.
- d) **Testing:** This phase is also known as verification and validation which is a process for checking that a software solution meets the original requirements and specifications and that it accomplishes its intended purpose. Verification is nothing but the process of evaluating any software that helps to determine it meets software specifications and that fulfil its purpose or not and validation is nothing but the process of evaluating whether it satisfies specified requirements or not. By the inputs from the system design the system is first developed in small programs that is called as units and these units are integrated in the next

phase. Each unit is developed and tested for its functionality that is called as Unit Testing. The testing phase is the outlet to perform debugging in which bugs and system glitches are found, corrected, and refined accordingly.

- e) **Deployment of system:** This is the next phase when the testing phase is completed. After the testing phase once the functional testing and non-functional testing is done for any product the product is deployed in the customer environment or it is released into the market.
- f) **Maintenance:** This is the last phase in software development. Maintenance is nothing but the process of modifying a software solution after delivery and deployment. It is used for various purpose like to refine output, correct errors, and improve performance and quality. Additional maintenance activities can be performed in maintenance phase. There are like adapting software to its environment, accommodating new user requirements, and increasing software reliability. There are many types of issues which come up in the client environment. So for fix these kinds of issues patches are released. For enhancing any products some better versions are released. Maintenance is done for delivering these changes in the customer environment.

## **ADVANTAGES OF WATERFALL MODEL**

- 1) This model is very simple and easy to understand and we can easily use this model for software development.
- 2) This model is easy to manage due to the rigidity of the model and each phase has specific deliverables and a review process.
- 3) All the phases are processed and completed one at a time.
- 4) This model works well for smaller projects where the requirements are very well understood.
- 5) This model clearly defined all the stages for software development.
- 6) It is easy to arrange tasks in waterfall model.
- 7) All the process and results are well documented in the waterfall models.

## **DISADVANTAGES OF WATERFALL MODEL**

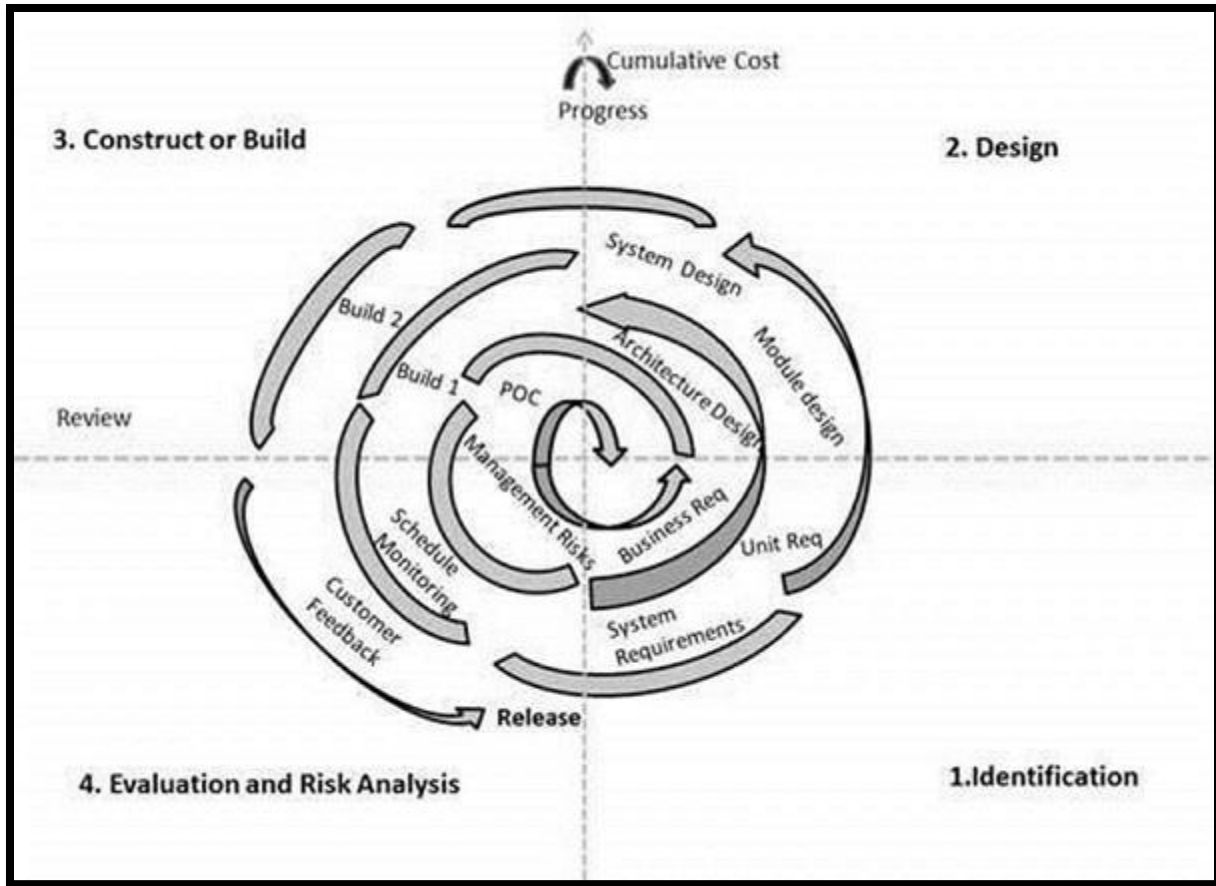
- 1) There is no working software is produced until late during the life cycle of the model.
- 2) This model contains the high amounts of risk and uncertainty.
- 3) The waterfall model is not a good model for complex and object-oriented projects.
- 4) The waterfall model is poor model for long and ongoing projects.
- 5) This model is not suitable for the projects where requirements are at a moderate to high risk of changing that is why risk and uncertainty is high with this process model.
- 6) It is difficult to measure progress within stages in the waterfall model.
- 7) This model cannot accommodate changing requirements.
- 8) In this model adjusting of scope during the life cycle can end a project.  
(Tutorialspoint, n.d.)



## 2) **SPIRAL MODEL**

The spiral model is also known as the spiral lifecycle model. This is a system development lifecycle model used in information technology. Spiral model is not so well known as other models. The spiral model is similar to the incremental development for a system with more emphasis places on risk analysis. The spiral model uses the various kinds of ideas of waterfall model. It includes iterative development with the systematic controlled and some other ideas. The Spiral model is nothing but a combination of various kind of models. We can say like iterative development process model or it is a sequential linear development model i.e. the waterfall model is basically very high emphasis on the risk analysis. The spiral contains the number of iterations and each iteration has incremental releases of any product or we can say incremental refinement of any product.

Spiral model can pretty costly to use and doesn't work well for a small projects. We can also say that Spiral model is risk-driven model. We can say this because the overall success of the projects highly depends on the phase called risks analysis phase. Risk analysis requires any specific expertise for each and every iteration. So Spiral model needs any special skills for reviewing the project and analyzing the project from time to time.(SDLC - Spiral Model, n.d.)



There are many software development life cycle phases in spiral model. Basically four main phases in spiral models. The whole development process passes through these stages. Each iteration is called spiral. The four main phases are Identification, Design, Construct or build, Evaluation and risk analysis.

- a) **Identification:** This is the first software development life cycle phase in spiral model. This phase deals with gathering requirements in the spiral. Identification phase consists of various parts like product matures, identification of system requirements, subsystem requirements and unit requirements completion. This phase also includes understanding the system requirements by continuous communication between the customer and the system analyst. The product is deployed and open in the market at the ending of spiral.
- b) **Design:** This is the second software development life cycle phase in spiral model after the identification phase. This phase contains the designing part. The design phase start with the conceptual design in the design phase of spiral. It also involves various designing part like architectural design, logical design of modules, physical product design and the final design in the subsequent spirals.
- c) **Construct or build:** This is the third software development life cycle phase in spiral model after the designing phase. The construct phase used to product the actual software product at the every spiral. In this phase when the product is decided then the design is being developed a proof of concept is developed in this phase to get a customer feedback. Subsequent spirals with higher requirements and design details a working model of the software is known as build. It is produced with version number. These builds are used for sending feedback to customer.
- d) **Evaluation and risk analysis:** This is the last software development life cycle phase in spiral model after the construct phase. This phase is mostly deals with the risk analysis. Risk analysis includes nothing but the identifying, estimating and monitoring the technical feasibility and management risks like schedule slippage and cost overrun. After testing the build in this phase and at the end of first iteration the customer evaluates the software and customer also provides feedback in this phase.

## **ADVANTAGES OF SPIRAL MODEL**

- 1) In this kind of software development lifecycle model changing requirements can be easily accommodated.
- 2) Spirals models allows extensive use of prototypes.
- 3) In Spiral software development lifecycle model requirements can be captured more accurately.
- 4) In Spiral software development lifecycle model users can see the system early.
- 5) Development can be divided into smaller parts in spiral model and the risky parts can be developed earlier because it is helps in better risk management in spiral models.

## **DISADVANTAGES OF SPIRAL MODEL**

- 1) In Spiral model management is more complex.
- 2) We may not know early about when the project is ended in Spiral model.
- 3) Spiral software development lifecycle model cannot suitable for small or low risk projects and it could be expensive for small projects.
- 4) Process is more complex in Spiral model compared to other SDLC models.
- 5) Sometimes there is chance for Spiral to may go on indefinitely.
- 6) Large number of intermediate stages requires excessive documentation in Spiral software development lifecycle model.

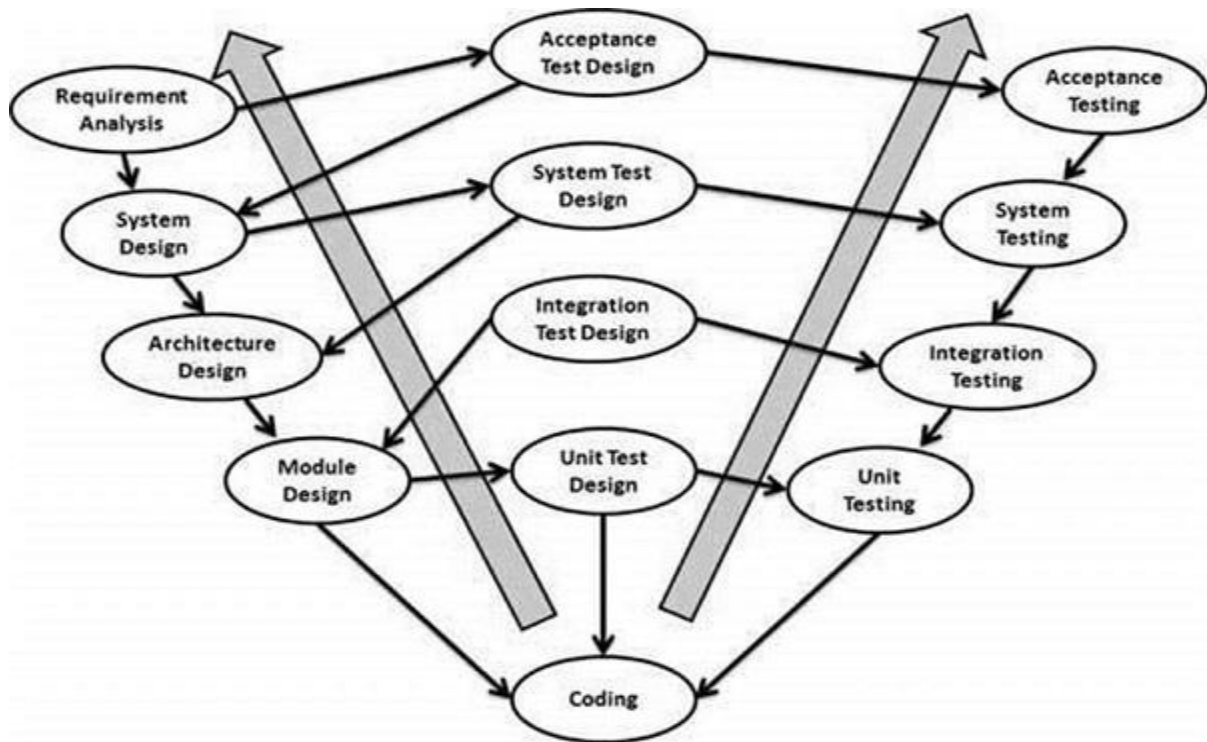
(SDLC - Spiral Model, n.d.)

### **3) V MODEL**

The V-model is also a type of a software development lifecycle model. The V-model is a software development lifecycle model where execution of processes happens in a sequential manner in a V-shape. This model is also known as Verification and Validation model. The testing activity is performed in the each phase of Software Testing Life Cycle phase. The V-model is nothing but the extension of the waterfall model. It is also based on the association of a testing phase for each corresponding development stage. So for each and every single phase in development cycle there is a directly associated testing phase.

In the first half of the model validation testing activity is integrated in each phase like review user requirements, system design documents and in the next half the verification testing activity is done. This is the highly-disciplined model and the next phase starts only after completion of the previous phase.

There are different types of parts in the V-model. They are like verification part, coding part and validation part. Typical V-model shows different parts like it shows software development activities on the left hand side of model and the actual testing phases can be performed on the right hand side of the model. (SDLC - V-Model, n.d.)



Here all phases of v-model as per below:

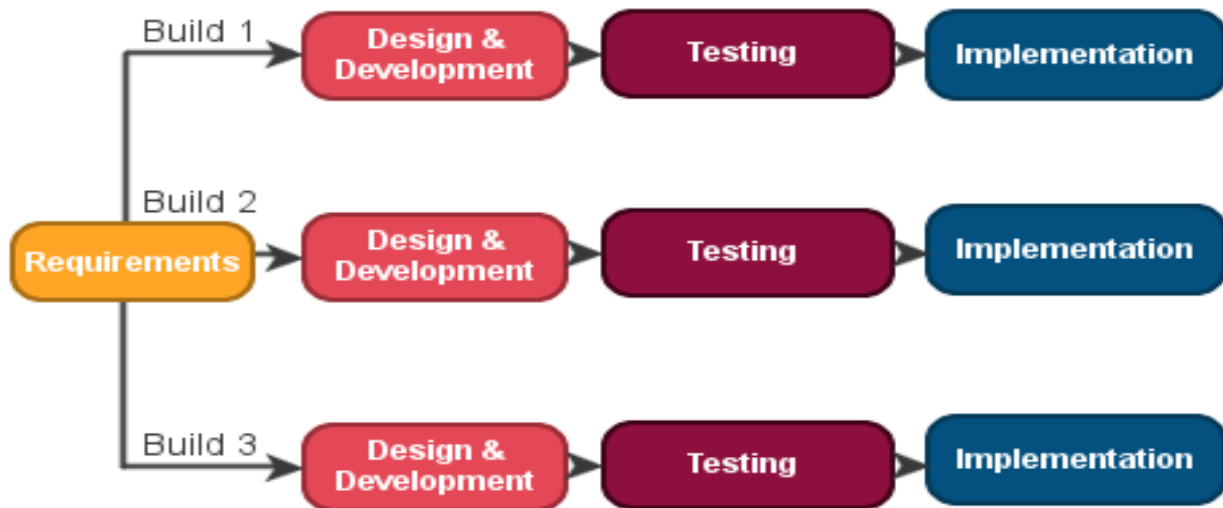
- a) **The Requirement phase:** a document show what the program is need to do after the gathered and analyzed the corresponding test processes is user acceptance testing.
- b) **The architecture design phase:** in this phase the architecture of program is designed and developed the different unit within program and established the relation unit and related test activity is integration testing.
- c) **System design phase:** in this phase the system divided in to small sub system with identified interface. Then convert in to more detail design and test activity is system testing.
- d) **Codding:** in this phase all component are coded and tested to verify if faithfully implemented detailed design.
- e) **Module design:** in this phase about the implementation of all unit specified. The whole design is divided in to data structure, algorithm and related testing is unit testing.

V-model is help full because it help to verification and validation in each and every stage of development. Every phase is able to test. This model is easy to understand and implement and use.

V-model have limitation also as it is not good to handle iteration or stage. Also it is hard to analysis risk analysis. In this model it hard to change in requirement in middle of project. (SDLC - V-Model, n.d.)

#### 4) Iterative model

Iterative model is also very important model in software development life cycle. In this model first step was divided whole set of requirement in different sub set as according to functionality. After this each iteration have less work to think on and develop and design. Please follow image below for more understanding.



As per image the requirement is going divided in to small set as per module or functionality. After that that small set of requirement is deliver to developer to design and development. Project manager assign small set of requirement to particular developer. After that developer design it and develop software as per condition of small set of requirement. Development phase is over here now it developed small functionality goes to tester to test it. After that tester follows all test steps and test case the developed functionality going to test and after everything is fine it working well as giver business requirement it going to final for implementation. In the end all small



functionality going to integration together and it all small modules and functionality connected to each other and converted into a software.

The iterative model is help full incase if we forget any functionality to development during development phase we can develop after and integrate with software. Incase if software is too big and it could take long time to get ready whole software we can develop necessary functionality on priority base. As we divide requirement into small sets we can develop that small set of requirement in parallel. We can track testing process development process and implementation. In case there will be need to change anything will be not that costly. Also in iterative model we can manage risk very easily. Iterative model is good for large and critical project.

On other side, iterative model required more resources. All resources should managed very well and schedule. In case you missed something you will come to know in the end of the project. This model need more skilled resources. This model is not suitable for small project. (SDLC - Iterative Model, n.d.)

# Software testing levels

Software development life cycle have final phase of whole process is deployment of software to the customer but before we release the software it needs to be undergoes from some testing levels. All the levels should be done in the fix orders. Each testing levels have their own different purpose and importance to software development process. Also some testing levels have sublevel too. But there are major testing levels are only four testing levels which are unit testing, integration testing, system integration testing and acceptance testing. There are other levels which can be in testing process add according to the testing objectives and prospective of customer.

## 1) Unit testing

The unit testing is first level of testing and it is very basic level of testing. This type of testing per specify to some particular units or module or group of codes and we need to makes sure that each part of unit is perfect. Unit testing test the basic functionality. Unit testing cannot testing able to test whole functionality of software but rather it do the component or modules which are working individually from each other.

Unit testing help to test internal design and internal logic, internal path and error handling. Unity testing is include only one type of testing is unit testing. Unit testing is perform by the software developer who develop the code to insure that the particular function is working as per requirement. Unit testing is the white box testing type.(Müller, Friedenber, Verma, & Veenendaal, 2007)

## 2) Integration testing

Integration testing is comes after unit test. Integration testing is all about the testing between component and software design. The integration testing is need to perform to verify the interface between unit or module with software design or system. Usually software components or unit or system or any modules are integrated between each other by iterative way or all together. This testing is done after unit test because we need to ensure that how components works individually and in integration testing we need to check

how it works together. The integration testing is ensure that the software is working in a well-organized and competent way.

There are few other type of testing that can be includes integration testing.

- Big bang testing
- Top down approach
- Bottom approach

The big band testing include the whole set or group of integrated module of components together at the same time. It will be ensure that everything which is integrated together and being tested at the same time can be find the root of problem.

The Top down approach test top most and important components first and it gradually move down and test the components as per priority level one after one.

The bottom up approach is same and opposite of top down approach.

In bottom up approach test first the lowest unit and it moved up by one by one. (Müller, Friedenber, Verma, & Veenendaal, 2007)

### **3) System testing**

System testing is test to ensure that the whole system is working as per requirement of not. This testing can be include the log in process, creating and ending actions required to use system or application, edit, delete, print, send the entries then log of means start to end process.

System testing can includes both test automation testing and manual testing. This testing is test the whole software at same time. To test system testing tester needs specific type of environment to test the application. System testing includes so many type of testing which can ensure to test the software from all criteria. System testing is most important type of testing because it test functional and non-functional also technical as well as business requirement of application. We can say this is the last level of testing because after this test user will test the application before they took handover. There is so many types of testing includes in this type of testing like regression testing, usability testing, performance testing, load testing,

functional testing, localization testing etc. (Müller, Friedenber, Verma, & Veenendaal, 2007)

#### **4) Acceptance testing**

Acceptance testing is the final stage of testing part of software development life cycle. It is also known as user acceptance testing. In this acceptance testing is to ensure that the software of application is ready to release or not. There is always many improvements in requirements and functionalities. It is very important to verify that is it meet the customer requirement and satisfy the business need before it release for people to use it. Also it is ensure about the functional and non-functional requirement. If there is any problem or no satisfaction form customer it will not go for real life use. It will go back to developer. Again they make changes in development again they follow all stage of testing and come back to the here again for acceptance testing. (Müller, Friedenber, Verma, & Veenendaal, 2007)

# Test Design Technique

There are different types of design technique are available to test particular kind of testing. Each of them are set for to ensure the particular types of errors or defects. Therefor all test design are classified in to two major types of test design technique Static and dynamic techniques.

## 1) Static Test design Technique

Static testing is ensure to test application manually or test with the help of tools. Static test should be done in the early testing phase of software development. In static testing usually testing of source code, function testing or requirement testing. Static test does not need computer to carry out the test result like walk through or reviewing. Almost all static test can done by the documents. Static testing should be done during verification process. Static testing have two classification manually and by tools. (Kaner, Falk, & Nguyen, 1993)

### a) Manual static test design technique

#### **Walk through**

Walk through is not that process which test and ensure by the testers. Tester can write the document and user can follow that documents and try to understand the test process to intend to get review and concept of understanding.

#### **Informal review**

Informal review is testing type where application is going to test without document and not in formal way. It just informal kind of testing to ensure to improve quality and have discussion to make it by best way. Informal review testing is test in primary stages and its involved the requirement documents and two member. As testing levels moves on another stages the team of inform review can be increase and also need more meeting.

#### **Technical review**

Technical review can carried out by the technical support team member.

Technical review test is less formal testing compare to inspection. It is usually test to get review without ant participation of management team.

**Audit**

Audit is the testing of the document of application it is usually done by the outside of person of company or development team. Audit is the proof for the customer of product that product is built as per standard development and controlled quality.

**Inspection**

An inspection is the formal kind of review test that carried out by the technical team member. In inspection test the requirement and product test that the product is built as per the requirement or not and if not try to find out bug and fix them on priority base. In inspection test document will be create for the defect and recommendation for improvement and take fallow up by date and time.

**Management review**

The management review is given by the company's management team. This team check the all work and check the each and every thing before deployment of product. This test is ensure for the customer satisfaction, cost value, performance of product and requirement fulfilled as per business plan.

**b) Static test design technique using by tools****Testing of coddng standards**

Analysis of coddng standards is consist standard of coddng. In this technique need to check the proper syntaxes of codes, rules of programming and code specification and validation of code. Compiler as tool help to verify this all criteria for coddng standards. Analysis of coddng should be in staring stages can save lots of effort, money and time.

**Testing of code metrics**

Code metrics is help to check the lines of code, cyclomatic number and nesting depth. This information is help to understand code easily lengthy code, code can be specify and help to maintain and understand code.

### **Testing of code structures**

Testing of code structures can help to give concept and logic about code to test by using tools. In code structures testing involves code control flow structure that set of instruction for execution. Also involves data flow structure provide understanding of data to use and update the coding. Also data structure provides us organization related data.

## **2) Dynamic test design technique**

Dynamic testing mainly includes execution of program on system. In this testing input data is need to insert to program and program give output as per input data. In this type of testing test criteria should be about memory use, performance, CPU usage and overall performance of program against output of program. This technique apply during validation process. Dynamic test design technique is divided into few parts are specification based, structures based and experience based. (Kaner, Falk, & Nguyen, 1993)

### **a) Specification based test design technique**

The specification based test design technique is the black box test design technique. In back box test design technique tester have permission only to test overview of functionality not the mechanism of functionality or coding. It includes technical specification, client requirement and Application functions etc. as per this criteria there are few methods specified as below.

#### **1) Boundary value testing:**

This is one of the most black box testing technique. In boundary value testing need to test input value which defines any boundary levels. Usually it is test to input values which are in between something like if any button or text box have input value is just 1 to 100 that means it will take from 2 to 99.

#### **2) Decision table**

Decision table test design technique use when there is input data is need to input in different box and with the different condition. Each input is connected with the output result. For example Gmail log in page we have to input email id then password then you need to click on log in button and you will get successful. So in case if you do not follow any action of one of the result will be none. Here three decision you need to complete is that email, password and the click on log in button.

### 3) **Stat transition testing**

The stat transition testing technique verify the condition from the status of transition. This test technique is use full when our system is implemented with different status with in them. Such as a bulb is connect with switch when is on it will give us light when you change the status from on to off it will stop working. So this test design technique is verify the stat transition between the workflow of application.

### 4) **Equivalence partitioning**

Equivalence test design technique is to verify the application when the input value are belongs to the same class. One classes are divided the test will follow same for all elements from same class. For example if there is any text box is implemented for number only. As per this test design technique we do not need to put each and every number and test. If one number is working then any number should working. Same as if any test box is implement for text character only then it should allow to input only text it will not allow any number or any other sign.

### 5) **Use case**

The use case testing is define as the method of testing the conditions form the use case. The test case is create on bases of output of the different business criteria and applications functionalities.

## **b) Structured based test design technique**

Structured base test design technique is white box type of testing. In white box testing tester need to test also the internal mechanism of application.in this type of testing tester have permission to test codding as well. As per this criteria some of more technique are specified below.

### 1) **Condition coverage**

Almost all of the test case designed as per white box test design technique derived by the condition coverage method. As per this test case are designed in manner that the output are executed automatically.

### 2) **Statement coverage**

Statement coverage method includes calculation of percentages of executable statement which are exercised by test suite.



### **3) Decision coverage**

In this decision coverage method, the test cases are designed in manner that the decision output executed automatically. It prove that each possible branch are from decision point implemented at least one time.

### **4) Multiple condition coverage**

Every combination of true and false for the condition related to a decision have to be tested by the multiple condition coverage technique.

## **c) Experience-based test design technique**

Experience base test design technique is define as there is no involvement of internal structure or external structure testing. This test design technique is only based on experience. Some of technique are specified below.

### **1) Exploratory testing**

This exploratory testing is run by the business analysis or some experts. They are test the application without any documents.

### **2) Fault attack testing**

The technique is most use full technique is experience base test design technique. This testing done by the tester with their previous project experience. They expect the defect in system as per their experience.

### **Things need to keep in mind while creating test case**

- To take pure logical coverage is to use decision table test design technique
- Boundary values and equivalence partitioning is the best test design technique to cover all input value.
- Modification and combination can be used for filed levels.

# Testing types

The various types of testing which are listed as below

## **Functional testing:**

Function testing is to ensure the main function of system or application that working well or not and it is executed properly or not. Functional testing should be follow as per requirement given by the client. The functional testing involves main operation, integration, data enter, data delete, data edit, use cases, user interface etc. (Software Testing by Gerald D. Everett and Raymond McLeod Jr.)

## **Non-functional testing:**

Non-functional testing are test which testing was left by the functional testing. Functional testing ensure mainly about business requirements. Non-functional testing mainly test the efficiency of the software that is it ready to hand over to client or public. Also it test that how fast your product is respond to command and how long it take to do action as per given command. Like performance testing, security testing, stress testing and usability testing etc.

## **Black box testing:**

Black box testing is define as while testing an application tester have no need to test internal structure or mechanism or any kind of coding. Black box testing principle is followed by acceptance testing and system testing as there are higher level of testing steps. This testing is just done by only general independent tester. While this testing no any kind of programing or implementation knowledge required to do this testing. In this testing only test case provided to ensure requirement specification.

## **White box:**

white box testing is define as while testing an application tester need to test internal structure or mechanism or any kind of coding. White box testing principle is followed by unit testing and integration testing as there are higher level of testing steps. This testing is just done by only general independent developer. While this testing need related kind of programing or implementation knowledge required to do this testing. In this testing only test case provided to ensure all detail design and business plan specification.

### **Gray box testing:**

Gray box testing is the combination of black box testing and white box testing. As black box testing no need to test internal structure and white box testing need to test internal structure but in gray box need to test partially internal structure. This testing need to access internal data exchange and design algorithm to create test case. In gray box testing tester have limited access to test internal structure but not in detailed.

### **Smoke testing**

Smoke test is apply to ensure to check that the application or software to is able or ready to testing for further testing processes. We can say that smoke testing is that initial type of testing. Smoke testing actually come from hardware side. This testing is check with hardware and when it switch on its catch fire or not. That's why it is known as smoke testing. To test smoke testing we need to create some basic test case. These test case need to execute in initial base to check that functionality is working normal or not and then application pass to test further testing procedure. Its help to check the integrated module issues on initial base so we can save time and money. Tester can fix those issue and avoid major issues in features. (Software Testing by Gerald D. Everett and Raymond McLeod Jr.)

### **Sanity testing**

After smoke testing application is test that stable enough to for further testing. We got application built that is have small or minor defect. After this defect fixed we need to run that sanity test to check that what we functionality had planned is working well. Actually sanity testing is done instead of whole regression testing. Sanity testing can save time and money in case our planned application got failed. Sanity test introduce the regression testing. After regression testing done we need to run sanity test to check that what we have change in system and code that's not breaking the main functionality of application. Sanity testing is apply in last phase of software development life cycle.

### **Positive testing**

Positive testing define as the test is tester with the valid test data. As per we know that testing is fully depend on test case there we need to enter test data to test application. Here we need to test only in positive way like we just need to test our application with the valid data only and need to check that is it behave as per expected output. Only positive scenario can test for application. In positive testing result always is positive.

## **Negative testing**

Negative testing define as the test is tester with the invalid test data. As per we know that testing is fully depend on test case there we need to enter test data to test application. Here we need to test only in negative way like we just need to test our application with the invalid data only and need to check that is it behave as per expected output. Only negative scenario can test for application. In negative testing result always is negative.

## **Functional testing**

Functional testing is one the most important testing in test processed. Functional testing is apply to ensure that developed application is built as per your business requirements. The functionality developed as per business requirement it should be functionally worked as it should be. The testing of this all major functionality, classes and module that it executed as per expected result is called functional testing.

## **Globalization testing**

The globalization testing is testing of geographic location and cultural environment. Our application should be run on different place with different language. Suppose we have created application to use not only particular place but for worldwide use. So if I run that application on Czech Republic it should load with Czech language and in case currency involve with any operation it should be in Czech crown automatically. It works on geographic location and cultural environment and this kind of testing known as globalization testing.

## **Alpha testing**

Alpha testing is the last phase testing before beta testing. Alpha testing is ensure to test defects and bug before product is going to deliver to the customer. Alpha testing is done by the developers. In this testing developer changes just minor thing because the main testing are already done before the alpha testing. For this type of testing we need to create virtual environment.

## **Beta testing**

Beta testing is last phase of testing. Usually it is done by be end user or customer. Beta testing is ensure to test that the product is behave as per given business requirement or not. Once the

testing is done and it meets the business requirement and the end user is satisfied after testing and they accept our developed product. That means best testing is pass.

### **Acceptance testing**

An acceptance testing is ensure to check that the application is flow to end to end. An acceptance testing is done by the end use. In acceptance testing is similar to beta testing. In acceptance testing client check that the product is meet the requirement and features and functionality is working well. After this acceptance testing done client accept the product.

### **Ad-hoc testing**

Ad-hoc testing is define as the testing process where no any data, reference or any documentation needed to test any application. Application or software is just need to test on general data or input and just need to check that the functionality is working or not. In this type of testing the bug is finding by any one while using the application of software. It is really very hard to find out the bug without any documentation about it but while testing Ad\_hoc testing we can find out bug which missed by test cases.

### **Accessibility testing**

Accessibility testing is define as the testing process where tester check the application that application is accessible for disable or not. In this type of testing tester need to check that application is use full and easy to get it like in case some people have some problem with eyes, color blind, bling, old people or mentally not good. Here they check the size of front that enough to read and contrast and brightness for whom who have eyes problem.

### **Back end testing**

When we enter the data in any application or software there is always database to save all store information. When we enter the data from application to database such kind of SQL server or MS SQL data is stored as per given instruction. In back end testing we do not need to check what application is doing. Here in this type of testing tester have some special kind of access to this database and he need to check by enter some query or some coddng and need to check that information is stored correctly or not.

## Compatibility testing

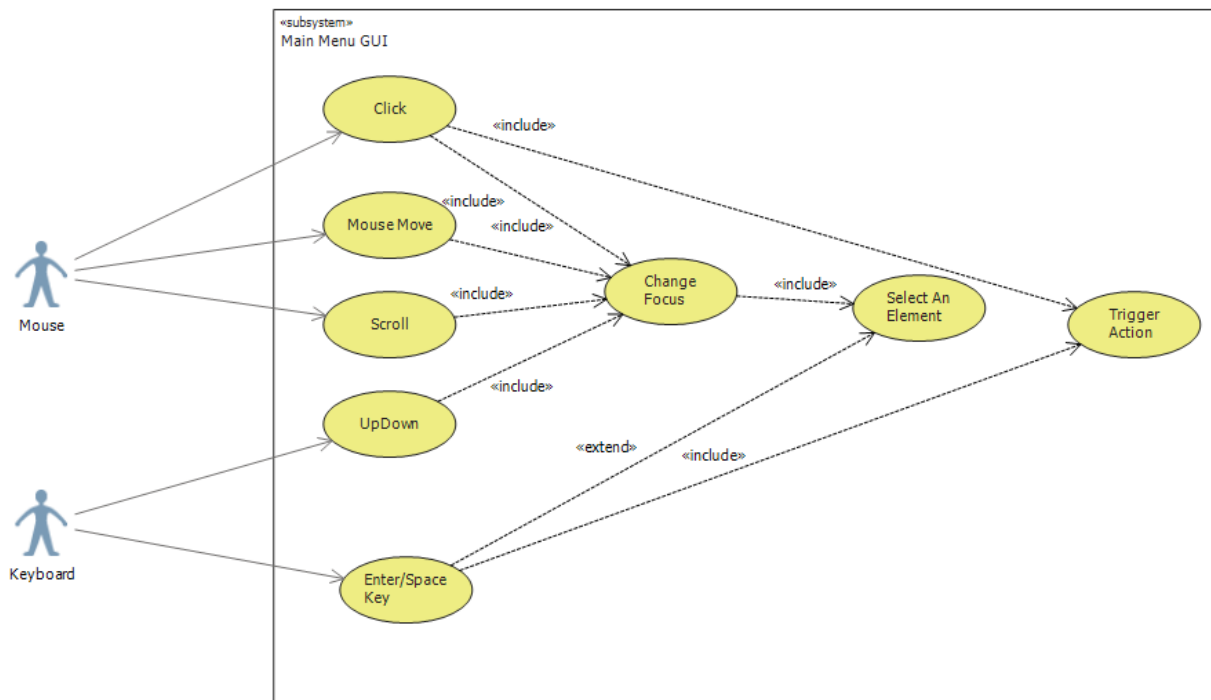
The compatibility testing is done by the testing team. In this kind of testing tester are test developed software in different environment like with different operating system, database or different hardware that is working well or not.

## Component testing

Component testing is need to test after unit testing. After done with unit testing all unit connected to each other and run a functionality. In component testing we need to test that component that work well or not after unit are connected to each other.

## Graphical user interface testing

The graphical user interface testing process ensure that the GUI is built are per given business requirement or not. In GUI testing tester are testing the button, check box and test box etc. in SDLC all information is given before the development starts and after that they developer that design of software. GUI testing also involves the size of words, spelling, color, table, alignment and contend and their place on screen.



## **Integration testing**

Integration testing is the one of the most important type of testing. In integration testing tester test the different module or different system or software to hardware working well when they are connected to each other. Integration testing test to the two different functionality react as it should be when they are connected to each other.

## **Load testing**

Load testing is one of the most use full non-functional testing. Load testing is ensure to check the capacity of software. Load testing will check how many user can use software as same and how software is behave while high rush and also check what is the difference comes in software performance. To test this type of testing tester need to use test tools like web load or jmeter etc.

## **Performance testing**

Performance testing is define as in this type of testing process tester need to check that software is respond as per fixed time or test of requirement set as per performance criteria.

## **Recovery testing**

The recovery testing is define as the testing done to verify that how application behave in disaster and crashes. In other words testing process verify that application or software take how much time to get back in stable stage after any crashes.

## **Regression testing**

The regression testing is define as the testing need to done after modification. After fix the founded bugs to check that the fixed bug effects any other functionality or not. To verify that the all functionality working well after the bugs fixed the regression testing need to apply. In regression testing need to test whole application as part to testing so usually regression testing is done by the automation tools.

## **Security testing**

Security testing is the part where tester check the authorization part of application. First in this testing tester team check that is it secure from unauthorized user or not. Second tester team check the application form hacking way and after that check that it is secure form any virus or not also it test form internal and external threat.

## **Usability testing**

Usability check testing is known as system navigation testing. In this testing application testing on based of it understanding level of user. How the application us built to know itself by user and check for that is it user-friendly or not. This testing is specially done for the new user that to check is it easy to understand or not. (Software Testing by Gerald D. Everett and Raymond McLeod Jr.)



# Test estimation methods

## Estimation

The test estimation method is one the management activity which can help to estimate how long a task or work would take to finished. The test estimation methods are one of the most and important part of test management. In test estimation process need to estimate time, resources, human skills and cost.

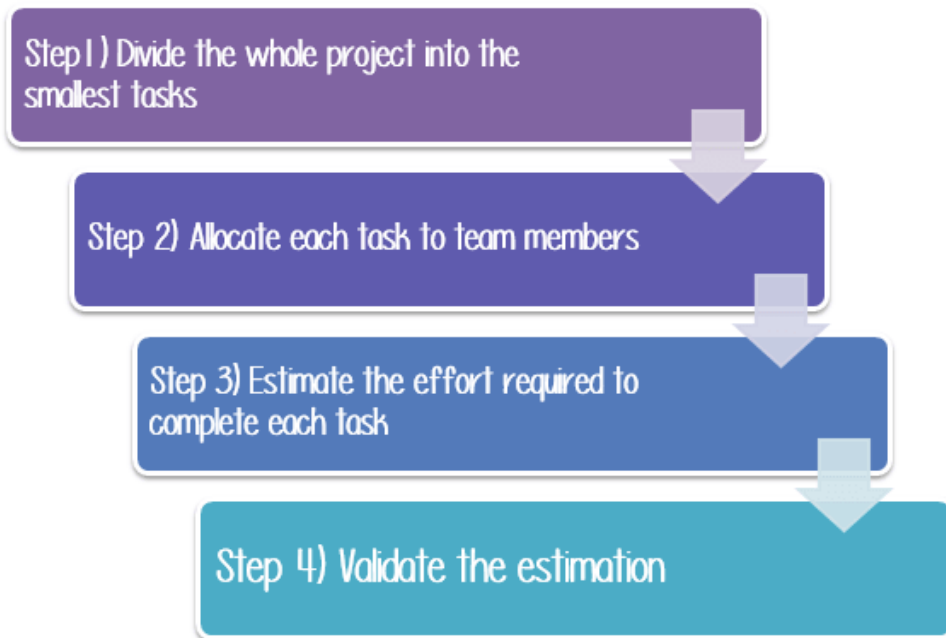


Resources are need to finished any task or project. Here in testing they can be people, things, money and facilities etc. time is the main part of the estimation methods as per we need to deliver the software on deadlines. As human skills they can be experience and skill. If there is high skilled and experience staff doing work on project they can finished early than junior and low skilled staff. Cost is the main aspect to develop anything here in testing we can say how much money need to spend to finished particular task.

**There are few estimation methods are as per below:**

### **Work break down method test estimation method**

The work break down method is the easiest method. In this method the whole method is break in to four major part as divide the whole task in to small parts, allocate particular task to in particular member, estimate the effort required to finished each task and last one is calculate the estimation.



As per first part the whole project is going to be divided into small parts after that small parts are going to be divided into modules after that module is going to be divided into sub-modules and that sub-module is going to be divided into functionality or classes. After that in the second part, as a particular task is allocated to particular members, so here the first part is to check the software requirements part, which is given to the business analyst team for analysis. After that it will go to design and development to developers. After developers finish their task, it goes to the testing team and they create test cases as per the given requirements and execute them and report a bug. In the third part, the project manager asks every one of the business analysts, developers, and testers about their work and how much time it takes to finish their work, and then it is finally validated to get the final answer that when the software will be ready.

### Three point test estimation method

Three point estimation is also one of the techniques to estimate the task. The three point's estimation uses the three types of points of estimation. Here they are as per below

- a = the best case of estimation
- m = the most likely estimation
- b = the worst case estimation

$$E = (a + 4m + b) / 6$$

$$SD = (b - a) / 6$$

Here E is the weighted average which will give you average answer to estimation about the project. SD will measure the variability and uncertainty in estimation.

(Software Test Estimation Techniques: Step By Step Guide, n.d.)

### **Planning poker test estimation method**

Planning poker is the one of the most important estimation technique used for agile based software development life cycle. Planning poker is estimate as per user stories derived by planning poker. The whole team of scrum involve and find out the result after play poker.

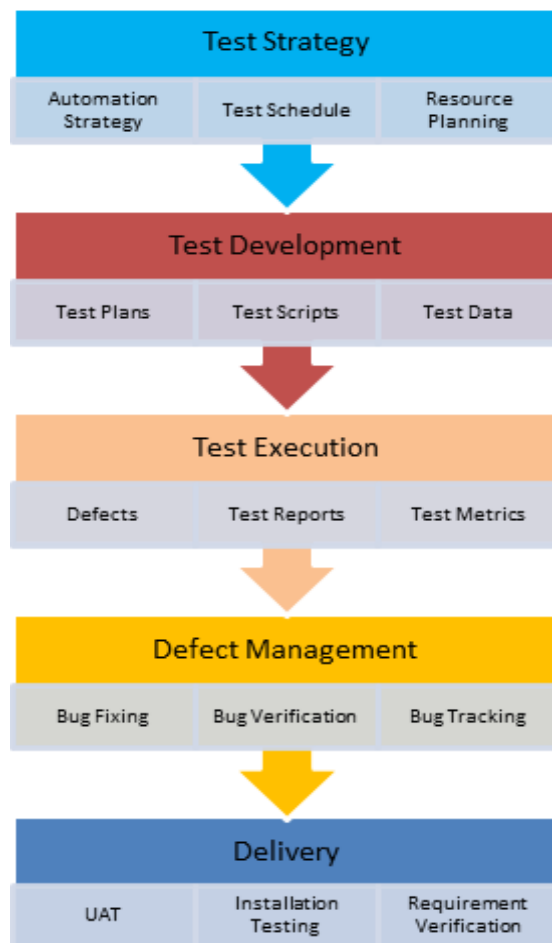
Planning poker is played with a deck of card where they used card as Fibonacci number as 1, 2, 3, 5, 8, 13, 21, 34, etc. this number show story points. In this game every team member take part and every team member have deck of cards. There is every team member have one member selected as moderator. This moderator read the user story and if any member have question they can ask the question about that user story then the product owner answer and explain about it. Every team member is separately selected a card which indicate there estimate. After that every show their estimation by showing their cards number and everyone can see each and every one number that how they estimated a story. And they validate every estimate and if there is huge difference between two team members then they need to explain why they choose such kind of estimation. After got conclusion and satisfactory answer they decide to finish that should in that validated time. Planning poker is use full when some time there is same type of user stories repeated in meeting. At that moment they do not need to repeat and discuss on such kind of story and they can easily estimate that story using past experience. (Estimation Techniques - Planning Poker, n.d.)

# Test strategy

The test strategy is a detail description of testing approach. The test strategy is most important document which have all answer for testing team how to get done with task. To develop test strategy document very tester develop skill with experience. The possibilities of missing any test activity is low when there is proper test strategy in documented. Test strategy should be discussed with whole team so team knows their responsibility. Test strategy include test approach, test environment, test tools, test plan, risk and issues and test objective.

## Test strategy document

For every company there is their own rule and regulation. They have their unique priority for develop software. So no one should follow the same test strategy for every project. Always keep in mind that test strategy should be develop after following requirement.



### **Step 1: scope**

In this step need to review the document and approve the document. The all testing process should follow by the schedule.

### **Step 2: Test approach**

In this step need to follow the testing procedure. Check that all type of testing level must be done. Every team member should follow the document as per schedule, responsibility and allocated task. Also need to follow non-functional type of testing like load performance etc. as per bug life cycle need to follow all stag as per status as defect after re-testing after defect triage after regression testing and sign off.

### **Step 3: test environment**

In this step need to fulfilled details about required environment for different environment. Also detailed about test data and restore strategy.

### **Step 4: testing tools**

In this step details about test management tools and automation testing.

### **Step 5: release control**

In this step details are mention about the product release. Test execution is should be taken care as per test plan and for all modification release.

### **Step 6: Risk analysis**

In this step there are estimated risk information details and mention.

### **Step 7: Review and approvals**

This is the last step in this step details about reviewed of all these activities and approved by the business team, management team and developer team. All information of changes in review should be marked at the document with assigned date, person and remarks. (How to create Test Strategy Document, n.d.)

# Test documentation

## Test plan

Test plan is one of the most important documents of software development life cycle. Test plan is detailed document about whole testing process. Test plan includes test strategy, test objective, planning, resources, hardware requirement, software requirement, testing schedule, testing deliverable and test estimation. Test plan is helping a lot in testing process and its play a vital role in as a software blue print. Test plan is create and control by the test manager or test lead. Test plan helping the test team as well as it is help full to people who are outside like developer and client to know about the testing processes. Test plan helping to follow step by step process to test team. Test plan main aspects are test estimation, test strategy and scope and it will be also use full in next project too.

Test plan need to follow some of important steps in test plan creation. In test management process as first step that need to analysis of product. In analysis of product test manager will check that developing product is about what. What it will be used for? Who will be use this software and how that developer developed software?

Second phase is the about test strategy. Test strategy is very difficult part of document. In this phase document create about testing objectives and how to test them in easy and sort time also about resources and cost of testing. Test strategy document include about scope of testing, testing type, risk issues and test logistics.

Third phase is about the test objectives. In this phase the details about the testing features and status of priority and functionality and test execution and which feature and functionality need to test as per priority.

Fourth phase is about test criteria. There is two criteria suspension criteria and exit criteria. If suspension criteria met in testing the test cycle will be suspended until the criteria are resolved.

(Kaner, Falk, & Nguyen, 1993)

Test plan ID	
References	
Description of test plan	
Test product	
Features to be tested	
Features to be not tested	
Approach	
Pass/fail criteria	
Suspension criteria	
Test deliverable	
Testing task	
Environment requirement	
Responsibilities	
Resource and training	
Schedule	
Risk	
Approval	

### **Test Case**

Test case is test process document about testing software. Test case is document which is create by the tester as per given business requirements for particular functionality. Test case have written step to follow how to test any functionality of class or module. As per test case tester is follow step as written in side and in case application is work as per giver step and result if actual result is same as expected result written in test case test is marked as pass in case it is not work as expected result test case if marked as fail. While creating a test case also help to find out the missing requirement in given requirement.

While write test case tester should be aware about that all positive and negative scenario are covered. Test case also follow and developed as per test design technique. Test case should be in simple language and easy to understand so any one can test by using test case. In test case only one thing should be test in one test case so there is no overlap or complicated test case. Test case should not have extra step except mandatory steps.(Kaner, Falk, & Nguyen, 1993)

## Test case template

test suit id	The Id of suit which the test case is belong to
test case id	The id of test case
TC description	Summary of the test case
Requirement	Requirement related test case
precondition	Condition should be fulfilled prior base to follow test
Test procedure	Test to executed test
Test data	Input value to fill in
Expected result	Expected result of developed functionality
Actual result	Real result of developed functionality
Notes	Remark or note or comments
Developed by	The test case is developed by whom
Developed date	The test case is developed date
Executed by	The test case tested by whom
Executed date	The test case tested date
Test environment	Test case testing needed environment

## Test case sample

**Test case ID:**

**Test Designed by:**

**Test title:**

**Test Designed date:**

**Pre-condition:**

**Dependencies:**

step	Test steps	Test data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1						
2						
3						



## **Bug report**

A bug report is the detailed report about the error or defect found in the software while testing it. Bug report is usually reported by the tester or testing team and assigned to the related developer or test team leader. The bug report is the document which help developer that to know about where they made mistake while development of software. Testers are should be more focused while creating test case and studying business requirement so developers have more specification in development of software and there will be less priorities to find a bug in the system.

## **Bug report sample**

Bug name	
Bug id	
Area path	
Build number	
Severity	
Priority	
Assigned to	
Assigned by	
Assigned date	
Reason	
Status	
Environment	
Description	
Step to reproduce	
Expected result	
Actual result	
Recommendation	

## **Test summery report**

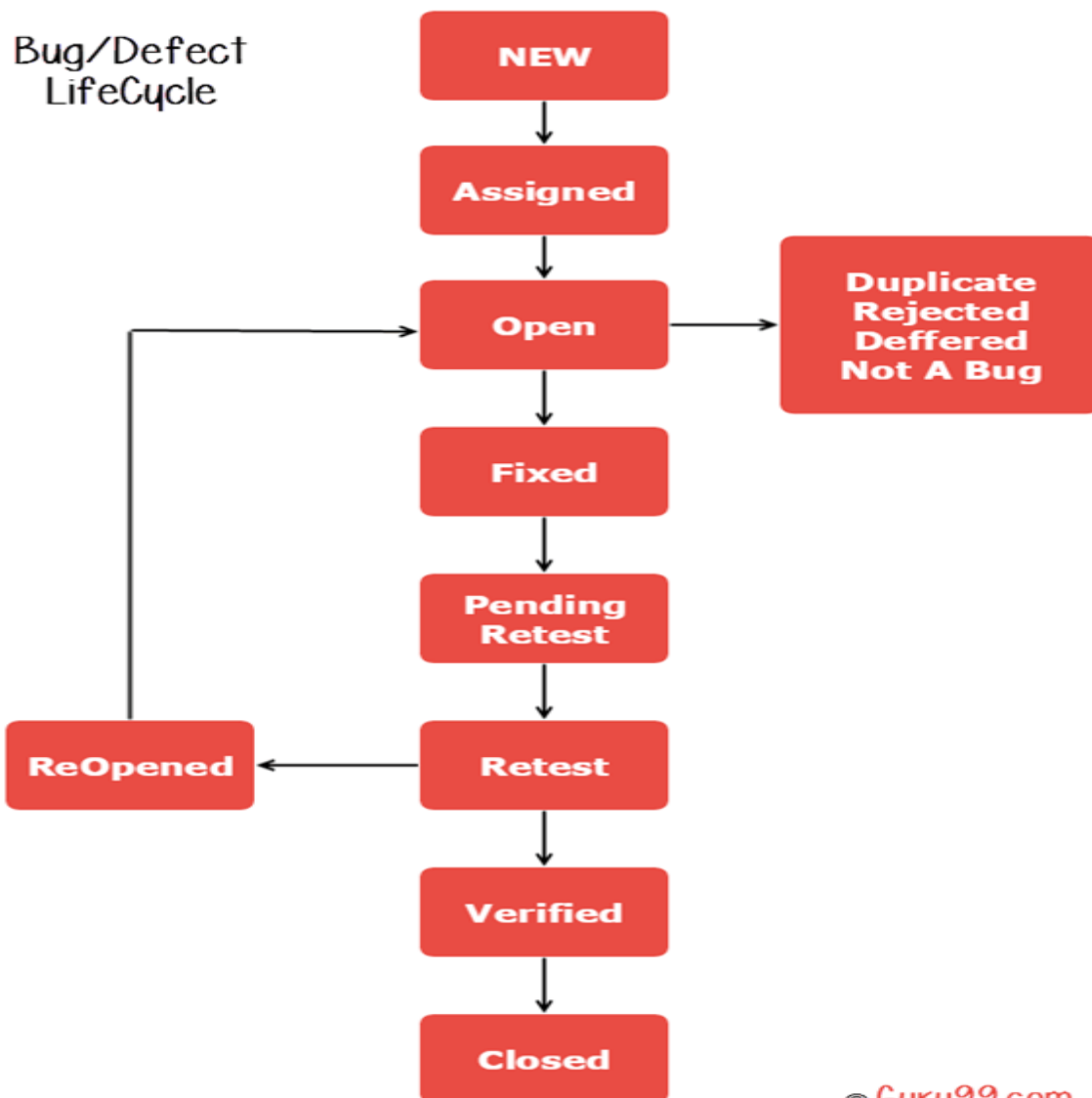
The test summery report is also one of the most important document which is create at the end of the testing process. The intention of this test summery report is to provide the details about the performed testing activities and details on the tested software. Test summery report is generally created by senior tester and hand over to product owner of test manager or leader. Test summery report helping use to estimate about the testing processes easily. The report also provide the testing process description, environment information, preconditions and comparison between test result and test approaches.(Kaner, Falk, & Nguyen, 1993)

**Test report sample**

Purpose of document:								
Purpose of document:								
Application overview:								
Testing scope: In scope:  Out of scope								
Metrics	Total TC	Planned TC	Executed TC	Passed TC	Failed TC	Defect Identified	Defect Status	Defect Severity
			0	0	0	0	0	0
Types of Testing Performed :								
Test Environment & Tools:								
Lesson Learned :								
Recommendations:								
Best Practice:								
Exit Criteria :								
Conclusion:								

## Bug life cycle

The bug life cycle is define as the set of states bug passes through found to fixation. Bug life cycle is deferent for each and every project. Upper image of bug life cycle is cover all the states bug need to go through. (Kaner, Software negligence & testing coverage, 1996)



Here are the all stages are with the description.

**New:** when the bug is found it is in the state of new.

**Assigned:** when bug found by the tester need to report the bug in bug tracking tool. After test manager or lead approve the reported bug and assigned it to developer.

**Open:** when bug is assigned to developer after developer open it and start working on it to fix.

**Fixed:** after the bug open analysis and fixed. Developer mark state as fixed.

Pending retest: after bug got fixed by developer it need to again goes to tester testing.

**Retest:** when tester retest the bug and change the state of bug as retest.

**Verified:** after testing again bug if tester do not found bug again and it fixed perfectly by developer it is state as verified

**Closed:** if there is no more bug found and but is in verified state it state as closed.

**Reopen:** after testing retest again bug if tester found bug again and it not fixed perfectly by developer it will state as reopen.

**Duplicate:** if bug is found repeatedly it is states as duplicate.

**Rejected:** once bug reached developer but bug is not creating any bad impact on system it state as rejected.

**Deferred:** if bug have not priority to fix on the emergency base than it can be state as deferred.

**Not a bug:** if developer found it is not genuine bug than its mark as not a bug.

## Automation testing:

The software test automation is make the use of special test tools to control execution of testing and test the actual result with the expected result. Usually like regression test need to automation because it need to repeat whole testing again and again.

The automation test tools are not only use full for GUI testing, performance testing, load testing, installation, uninstallation, data set up generation and defect logging etc.

There are criteria to choose the tools in automation. The criteria are such as data driven capacity, email notification, version friendly and platform independently etc.

- There are major four types of frameworks are adopted while automation testing are such as Data Driven Automation Framework, Keyword Driven Automation Framework, Modular Automation Framework, Hybrid Automation Framework. (Software Testing by Gerald D. Everett and Raymond McLeod Jr.)

## Automation testing tools:

Testing tools available are as per below to test function testing:

NO.	Tool	Vendor
1	Auto It	It an open source
2	Selenium	It an open source
3	Coded UI	Microsoft
4	Quick test professional	HP
5	Rational Robot	IBM

Testing tools are available are as per below to test non-functional testing:

NO.	Tool	Vendor
1	Jmeter	Apache
2	Burn Suite	Portswigger
3	Load Runner	HP
4	Acunetix	Acunetix

# Appendix

Here in this section is practical work of the thesis methodology about the thesis. Thesis methodology how work in real way it is show by this section of thesis. Here we will see by sample of small requirement and how its work by follow software testing test design technique and also documentation. (It is just an example)

## **Requirement:**

**Test subject: web site 'triangle'.**

## **UI Description.**

- 3 fields to enter length of triangle side.
- Entered length should be in centimeters.
- Points between values will be taken as invalid only for field 1 and 2, for filed three will be taken as valid.
- Maximum input for all fields is 4 digits.
- A button that should be active only if all fields are filled in. Click action should trigger creation of picture of triangle using entered length side by user.
- Non editable field where a picture of triangle is displayed

As per given requirement we will create test case and also show a bug report about it. It is not real application. It is just assumed. Test case are as per below.

## Test case template for web site “triangle”

**Test case ID:** 01

**Test Designed by:** Ankit Vaniya

**Test title:** valid input for all fields

**Test Designed date:** 18.07.2017

**Pre-condition:** website “triangle” launched

**Dependencies:**

step	Test steps	Test data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Enter length in filed 1	1	1 should be display in filed 1			
2	Enter length in filed 2	1	1 should be display in filed 2			
3	Enter length in filed 3	1.41	1.41 should be display in field 3			
4	Check the button		Button is activated			
5	Click action on button		Picture of triangle is created in field 4			

**Test case ID:** 02

**Test Designed by:** Ankit Vaniya

**Test title:** null input for field1 and valid input for fields 2 and 3

**Test Designed date:** 18.07.2017

**Pre-condition:** website “triangle” launched

**Dependencies:**

step	Test steps	Test data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Enter length in filed 1	Null	Nothing should be display in filed 1 and highlighted with red			

			color			
2	Enter length in filed 2	5	5 should be display in filed 2			
3	Enter length in filed 3	5	5 should be display in field 3			
4	Check the button		Button is not activated			
5	Click action on button is unable		Field 4 is empty			

**Test case ID:** 03

**Test Designed by:** Ankit Vaniya

**Test title:** null input for field 2 and valid input for all fields 1 and 3 **Test Designed date:** 18.07.2017

**Pre-condition:** website “triangle” launched

**Dependencies:**

step	Test steps	Test data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Enter length in filed 1	5	5 should be display in filed 1			
2	Enter length in filed 2		Nothing should be display in filed 2 and highlighted with red color			
3	Enter length in filed 3	5	5 should be display in field 3			
4	Check the button		Button is not activated			
5	Click action on button is unable		Field 4 is empty			

**Test case ID:** 04

**Test Designed by:** Ankit Vaniya

**Test title:** null input for field 3 and valid input for all fields 1 and 2 **Test Designed date:** 18.07.2017

**Pre-condition:** website “triangle” launched

**Dependencies:**



step	Test steps	Test data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Enter length in filed 1	5	5 should be display in filed 1			
2	Enter length in filed 2	5	5 should be display in filed 2			
3	Enter length in filed 3	null	Nothing should be display in filed 3 and highlighted with red color			
4	Check the button		Button is not activated			
5	Click action on button is unable		Field 4 is empty			

**Test case ID:** 05

**Test Designed by:** Ankit Vaniya

**Test title:** null input for all fields

**Test Designed date:** 18.07.2017

**Pre-condition:** website “triangle” launched

**Dependencies:**

step	Test steps	Test data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Enter length in filed 1	null	Nothing should be display in filed 1			
2	Enter length in filed 2	null	Nothing should be display in filed 2			
3	Enter length in filed 3	null	Nothing should be display in filed 3			
4	Check the button		Button is not activated			
5	Click action on button is unable		Field 4 is empty			

---

**Test case ID:** 06

**Test Designed by:** Ankit Vaniya

**Test title:** null input for field 1 and 2 valid input for all fields 3 **Test Designed date:** 18.07.2017

**Pre-condition:** website “triangle” launched

**Dependencies:**

step	Test steps	Test data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Enter length in filed 1	null	Nothing should be display in filed 3 and highlighted with red color			
2	Enter length in filed 2	null	Nothing should be display in filed 3 and highlighted with red color			
3	Enter length in filed 3	5	5 should be display in filed 3			
4	Check the button		Button is not activated			
5	Click action on button is unable		Field 4 is empty			

---

**Test case ID:** 07

**Test Designed by:** Ankit Vaniya

**Test title:** null input for field 3 and 2 valid input for all fields 1 **Test Designed date:** 18.07.2017

**Pre-condition:** website “triangle” launched

**Dependencies:**

step	Test steps	Test data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Enter length in filed 1	5	5 should be display in filed 3			
2	Enter length in filed 2	null				
3	Enter length in filed 3	null	Nothing should be display in filed 3 and highlighted with red			

			color			
4	Check the button		Button is not activated			
5	Click action on button is unable		Field 4 is empty			

---

**Test case ID:** 08

**Test Designed by:** Ankit Vaniya

**Test title:** null input for field 3 and 1 valid input for all fields Test 2 **Designed date:** 18.07.2017

**Pre-condition:** website “triangle” launched

**Dependencies:**

step	Test steps	Test data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Enter length in filed 1	null	5 should be display in filed 3			
2	Enter length in filed 2	5				
3	Enter length in filed 3	null	Nothing should be display in filed 3 and highlighted with red color			
4	Check the button		Button is not activated			
5	Click action on button is unable		Field 4 is empty			

---

**Test case ID:** 09

**Test Designed by:** Ankit Vaniya

**Test title:** minimum valid input for all fields

**Test Designed date:** 18.07.2017

**Pre-condition:** website “triangle” launched

**Dependencies:**

step	Test steps	Test data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Enter length in filed 1	1	1should be display in filed 1			
2	Enter length in filed 2	1	1 should be display in filed 2			
3	Enter length in filed 3	1.41	1.41 should be display in filed 3			
4	Check the button		Button is activated			
5	Click action on button		Picture of triangle is created in field 4			

**Test case ID:** 10

**Test Designed by:** Ankit Vaniya

**Test title:** minimum valid input for all fields

**Test Designed date:** 18.07.2017

**Pre-condition:** website “triangle” launched

**Dependencies:**

step	Test steps	Test data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Enter length in filed 1	1	1should be display in filed 1			
2	Enter length in filed 2	1	1 should be display in filed 2			
3	Enter length in filed 3	2	Nothing should be display in filed 2			
4	Check the button		Button is not activated			
5	Click action on button is dis able		Field 4 is empty			

**Test case ID:** 11

**Test Designed by:** Ankit Vaniya

**Test title:** invalid input for all field 1, value with minus sign in field 2 and character input for field 3

**Pre-condition:** website “triangle” launched

**Test Designed date:** 18.07.2017

**Dependencies:**

step	Test steps	Test data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Enter length in field 1	0	Nothing should be display in field 1 and highlighted with red color			
2	Enter length in field 2	-10	Nothing should be display in field 2 and highlighted with red color			
3	Enter length in field 3	Chr22	Nothing should be display in field 3 and highlighted with red color			
4	Check the button		Button is not activated			
5	Click action on button		field 4 is empty			

**Test case ID:** 12

**Test Designed by:** Ankit Vaniya

**Test title:** invalid input for all field 2, value with minus sign in field 3 and character input for field 1

**Test Designed date:** 18.07.2017

**Pre-condition:** website “triangle” launched

**Dependencies:**

step	Test steps	Test data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Enter length in filed 1	Hdfsh25	Nothing should be display in filed 1 and highlighted with red color			
2	Enter length in filed 2	0	Nothing should be display in filed 2 and highlighted with red color			
3	Enter length in filed 3	-10	Nothing should be display in filed 3 and highlighted with red color			
4	Check the button		Button is not activated			
5	Click action on button		field 4 is empty			

**Test case ID:** 13

**Test Designed by:** Ankit Vaniya

**Test title:** invalid input for all field 3, value with minus sigh in filed 1 and character input for field 2

**Test Designed date:** 18.07.2017

**Pre-condition:** website “triangle” launched

**Dependencies:**

step	Test steps	Test data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Enter length in filed 1	-121	Nothing should be display in filed 3 and highlighted with red color			
2	Enter length in filed 2	Gdsag21	Nothing should be display in filed 3 and highlighted with red			

			color			
3	Enter length in filed 3	0	Nothing should be display in filed 3 and highlighted with red color			
4	Check the button		Button is not activated			
5	Click action on button		field 4 is empty			

**Test case ID:** 14

**Test Designed by:** Ankit Vaniya

**Test title:** maximum valid input for all fields

**Test Designed date:** 18.07.2017

**Pre-condition:** website “triangle” launched

**Dependencies:**

step	Test steps	Test data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Enter length in filed 1	999	999 should be display in filed 1			
2	Enter length in filed 2	999	999 should be display in filed 2			
3	Enter length in filed 3	999	999 should be display in field 3			
4	Check the button		Button is activated			
5	Click action on button		Picture of triangle is created in field 4			

**Test case ID:** 15

**Test Designed by:** Ankit Vaniya

**Test title:** validation of element availability on screen

**Test Designed date:** 18.07.2017

**Pre-condition:** website “triangle” launched

**Dependencies:**

step	Test steps	Test data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
------	------------	-----------	-----------------	---------------	--------------------	-------

1	Verify that field 1 on screen on first position		Filed one is on first position			
2	Verify that field 2 on screen on first position		Filed two is on first position			
3	Verify that field 3 on third on first position		Filed three is on first position			
4	Check the button		Button should be unable to click			
5	Verify that filed 4 is on last position		Filed for should be empty			

**Test case ID:** 16

**Test Designed by:** Ankit Vaniya

**Test title:** verification of OS

**Test Designed date:** 18.07.2017

**Pre-condition:** website “triangle” launched

**Dependencies:**

Test case no.	Google chrome	Safari	Mozilla Firefox	Opera
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				
11				

**Test case ID:** 17

**Test Designed by:** Ankit Vaniya



**Test title:** input values with points for all fields

**Test Designed date:** 18.07.2017

**Pre-condition:** website “triangle” launched

**Dependencies:**

step	Test steps	Test data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Enter length in filed 1	1.2	Nothing should be display in filed 1			
2	Enter length in filed 2	1.5	Nothing should be display in filed 2			
3	Enter length in filed 3	1.41	1.41 should be display in filed 3			
4	Check the button		Button is not activated			
5	Click action on button is unable		Filed 3 is empty			

### **Sample Bug report for website “triangle”**

**Bug Name:** In website ‘triangle’ after filling length in all three field button is not going activated.

**Bug ID:** test case 01

**Area Path:**

**Build Number:** 01

**Severity:** high

**Priority:** high

**Assigned to:** Mariana

**Reported By:** Ankit Vaniya

**Reported On:** 02.08.2018

**Reason:** defect

**Status:** new

**Environment:** windows 10

**Description:** unable to create triangle when clicking on button.

**Steps To Reproduce:**

- 1) Launch the website “triangle”
- 2) Enter length in filed 1, 2 and 3.
- 3) Check if button is active.

**Expected result:** button should be activated after clicking the button picture of triangle should be create in filed 4.

**Actual result:** button is not activated after felling valid inputs.

## **Conclusion:**

This thesis have made an introduction into most important phase of the software development life cycle which is software testing. Software testing has a significant impact to the quality of the software application but sometimes underestimated it. This is why in this thesis, we described necessary criteria for testing a software as test design technique test step, test type and test estimation as well as to prepare documentation on testing like test case, test report, bug report and test plan.

This thesis consisted also a practical part showing an example of various user interface elements testing on the all-important software platforms (e.g Windows, iOS, Unix).

# Bibliography

*Estimation Techniques - Planning Poker.* (n.d.). Retrieved from tutorialspoint:

[https://www.tutorialspoint.com/estimation\\_techniques/estimation\\_techniques\\_planning\\_poker.htm](https://www.tutorialspoint.com/estimation_techniques/estimation_techniques_planning_poker.htm)

*How to create Test Strategy Document.* (n.d.). Retrieved from guru99:

<https://www.guru99.com/how-to-create-test-strategy-document.html>

Kaner, C. (1996). *Software negligence & testing coverage.*

Kaner, C., Falk, J., & Nguyen, H. Q. (1993). *Testing computer software 2nd Edition.*

Müller, T., Friedenber, D., Verma, R., & Veenendaal, E. (2007). *Foundations of Software Testing.* International Software Testing Qualifications Board. Retrieved March 30, 2018, from [http://www.sl-stb.org/docs/ISTQB\\_SyllabusCTFL2011.pdf](http://www.sl-stb.org/docs/ISTQB_SyllabusCTFL2011.pdf)

*SDLC - Iterative Model.* (n.d.). Retrieved from tutorialspoint:

[https://www.tutorialspoint.com/sdlc/sdlc\\_iterative\\_model.htm](https://www.tutorialspoint.com/sdlc/sdlc_iterative_model.htm)

*SDLC - Spiral Model.* (n.d.). Retrieved from Tutorialspoint:

[https://www.tutorialspoint.com/sdlc/sdlc\\_spiral\\_model.htm](https://www.tutorialspoint.com/sdlc/sdlc_spiral_model.htm)

*SDLC - V-Model.* (n.d.). Retrieved from tutorialspoint:

[https://www.tutorialspoint.com/sdlc/sdlc\\_v\\_model.htm](https://www.tutorialspoint.com/sdlc/sdlc_v_model.htm)

*Software Test Estimation Techniques: Step By Step Guide.* (n.d.). Retrieved from guru99:

<https://www.guru99.com/an-expert-view-on-test-estimation.html>

Tutorialspoint. (n.d.). *SDLC - Waterfall Model.* Retrieved from Tutorialspoint:

[https://www.tutorialspoint.com/sdlc/sdlc\\_waterfall\\_model.htm](https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm)

Michael Dyer, *The Cleanroom Approach to Quality Software Development*, Wiley, 1992.

M. Friedman and J. Voas, *Software Assessment: Reliability, Safety, Testability*, Wiley, 1995.

C. Kaner, J. Falk, and H.Q. Nguyen, *Testing Computer Software (2/e)*, Van Nostrand Reinhold, 1993.

Cem Kaner, "Software Negligence & Testing Coverage," in *Proceedings of STAR 96, (Software Quality Engineering, Jacksonville, FL)*, 1996.

Brian Marick, *The Craft of Software Testing*, Prentice Hall, 1995.

<https://www.codeproject.com/Tips/351122/What-is-software-testing-What-are-the-different-ty>

<http://istqbexamcertification.com/what-is-fundamental-test-process-in-software-testing/>

<http://istqbexamcertification.com/what-is-fundamental-test-process-in-software-testing/>

<https://www.utest.com/articles/seven-testing-principles>

<https://www.ijcsi.org/papers/IJCSI-11-2-2-120-123.pd>

<https://blog.testlodge.com/levels-of-testing/>

<https://www.invensis.net/blog/it/software-test-design-techniques-static-and-dynamic-testing/>