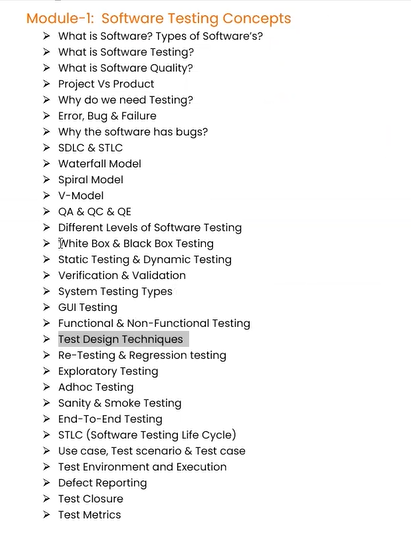
## Manual Testing, STLC, Agile, Jira, e-Commerce Project: <https://capgemini.udemy.com/course/learn-manual-software-testing-with-live-project-jira-tool/learn/lecture/13232928#overview> Instructor

[**Pavan Kumar**](https://capgemini.udemy.com/user/3ddc902a-47a4-48a3-b17b-d0e9160c1daa/)

****Tech Educator | Trainer | YouTuber  
 **Full-stack QA: Topics will cover**  
1) Manual Testing – As a tester we have known some basic concepts from manual testing like what

are different testing concepts, testing process, agile so these things we have to aware.  
   
2) Java programming – If we want to go for automation tester, we need to learn some programming

concepts so java is required.  
  
3) Selenium – It is an automation tool which is available in the market, it is very popular now a days in a market.  
 java and selenium are coming from automation part.  
  
4) Advanced concepts of automation testing is:  
 Maven, Jenkins, Git, GitHub, and Docker

5) We are going to do two different projects are:  
 1) Manual testing project  
 -> How to analyse the requirement, how to write the test plan, how to write the test cases, and how

to execute, how to find the defects etc. So, these things will the part of manual testing project.

2) Automation testing project

-> This is the topics is just like a full stack QA as a beginner, if only manual testing difficult to survive in

the industry so, we need to know automation testing.

1. **Manual Testing:**We are going to learning as a manual testing in three different modules are:
2. **Module - 1**: Software testing Concepts (Theory) what to test?

what is software testing?

What are the different levels of testing?

What are the different types of testing we have to conduct in our application?  
What is software? What is the different software are available

SDLC, STLC etc. we are going to learn some basic concepts from testing before conducting the testing we have to know what kind of testing are there and how are conduct those testing.

1. **Module - 2**: Testing project (Practical) How to test?

How to analyse a requirement?

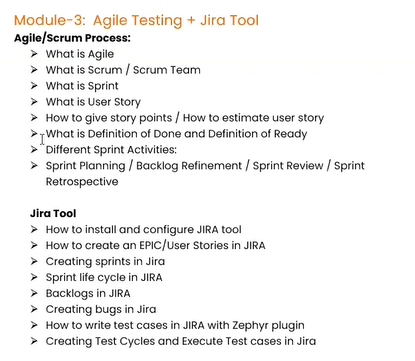
How to write the test plan?

How to write the test cases?

How to execute our test cases?

And we are going to learn about bug or defect life cycle, how to report the bugs

These concepts are purely practical part



1. **Module – 3**: Agile process (important tool is Jira)

Agile process is very important in most of the company will follow the agile process

How we will get the requirement How to write the user stories

How we can do the backlog plan or sprint plan

How we can interact within the project team scrum.

So, Jira is a basically a Agile test management tool or we can say Agile tool and in this particular application we can track everything in the software development from the beginning from the requirement to till we deliver to the software to customer everything we are tracking to this particular tool even testers and developers each and every activity test cases, test plans, execution status, reports everything we can managing in this tool.

Before Jira tool people will using excel sheet and world documents instead of, we are using Jira tool.

**SDLC:** Software development life cycle

It is a process used by software industry to design, develop and test software’s and finally deliver to the customer.

PPP: people, process, and product these three pillars are very important

**People** are very important to organization to developing the application to the customer. Without people there is no company.

**Process** We have to achieve something, we have to follow certain process and certain stages finally we can achieve the goal

**Product** Finally whatever we deliver to the customer that is called as product or application, if we want to deliver to the customer people should follows certain process then we can deliver to the product to customer

So, this SDLC is related to the process which will normally talking about the process which we have to follow to design and develop test the software finally we deliver the software to the customer.

So, software development will happen certain process is followed:

**Requirement analysis** -> **Design** -> **Development** -> **Testing** -> **Maintenance**

**Requirement analysis –** We have to collect and understand the requirement from the customer, so normally project manager involved in this phase.

**Design –** Once we get the requirements from the customerthe design phase what happens is the designers will design the software like UI part, navigation part etc.

**Development –** Software developer will develop the software by writing different type of programs by using different types of languages.

**Testing –** After development is completed before delivering a software to the customer, we have to test it we have conduct different types of testing like functional, non-functional, security and normally the people were involved in the testing phase we can call them as a tester for QA or QC all these people were involved in the testing phase.

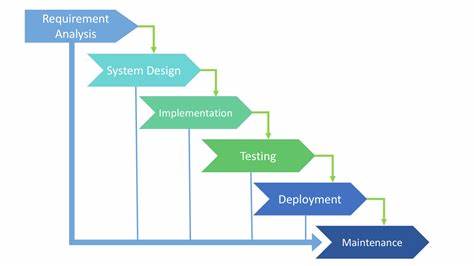
**Maintenance –** After that finally we deliver the software to the customer environment and maintenance will be start to the customer environment and start using the software.

So, these are the steps we need follow to deliver the quality product to the customer.

Company will design or they follow different models all this comes under SDLC only but the approach the way of following the phases is different. What are those models:

**Waterfall model** **spiral model** **V-model**

So, Prototype is also one of the SDLC model which is also available comes under between waterfall and spiral model.

**Waterfall model** – This is very old and traditional model, The first and initial model is started from waterfall model this is also called as **linear model.** Even though we are not following this model now a days in the company, but we should understand this previously how models are there what are the drawbacks we face, and we are implement the new models. So, all these models again come under SDLC only the approach is little different.  


**What is waterfall model:**

This is also one model comes under SDLC and whatever phases in the SDLC same phases will be there, but the approach will be different.

**Requirement analysis** -> **System design** -> **Implementation** -> **Testing** -> **Deployment** -> **Maintenance**

**Requirement analysis –** We collect the requirement, or we gathered the requirement form the customers and we will prepare certain document, so here the main importance part is documentation in the sense each and every activity will be document, normally talk to the customers are business people like project managers, product managers these peoples involved in the communication between customers and companies so they will collect the requirement from the customers and once get the requirements they prepare so kind of document that is call **SRS** **document** (Software requirement specification document) so in that document they will document every requirement from the customer so once the document is created then system design will started

**System design –** It means whatever the document they prepared during requirement analysis based on the document the designers will be there again the different team so those people will design the software and they will prepare some (high-level design) we can call it as **HLD** and **LLD** (low-level design) like basically they will divide the entire requirement into **HLM** (High level model) and **LLM** (Low level model) small pieces of software and finally will integrate them.

**Implementation –** Once the design part is completed then the developers will develop or implement the software based on the system design document, because as a developer want to develop a software need some input from where I get the input so based on the design, we have to develop the software.

**Testing –** After completed the development or implementation then the testing part will start, so in the testing part what will do testers will test the software after completion of development

**Deployment -** Once the testing is completed then deploy the software to the customer environment, it is nothing but will install the software in the customer environment.

**Maintenance –** Once the software is installed start using the software, so this is the maintenance.

**\*** Each and every face is having certain input and it will give some outcome

\* Suppose If I want to start a design until and unless the requirement analysis completed, I cannot start the design part, every phase is depending on the like maintenance is depends on the deployment and deployment is depends on the testing, testing is depending on the implementation, implementation is depending on the design part and system design is depends on the requirement analysis.

\* So, each phase is depending on the previous phase. This is the behaviour of the waterfall model

**What are the advantages and disadvantages:**

**Advantages:**

1) The quality of the product will be good – Because every face we have a detailed documentation so whatever activity we do we will document it. It will very clear to the who are all involved to the design testing and development face.

2) Since the requirement changes are not allowed, changes of finding bugs will be less.

3) Initial investment is less since the testers are hired to the later stages.

4) Preferred for small projects where requirements are freezed.

**Disadvantages:**

Based on the drawbacks or disadvantages the new models are coming into market

1) Requirement changes are not allowed

2) there is a defect in requirement that will be continued in later phases.

It means in the requirement analysis phase we have collected some requirements suppose if there is any problem any mistake in the requirement that will be again continued in the next phases.

3) Total investment is more, because time taking for rework on defect is time consuming which leads to high investment.

It means suppose in the testing phase we found some defects and when a found defects in a testing phase we have to correct the defects in the coding phase and again we have to correct at the design level again we have to correct at the requirement level. So, if we find any defects in any of those phases we have to go back to the previous phase and we need to modify the document accordingly.

4) Testing will start only after coding.

So, we have a one phase of testing the bugs chances will be more, if the bugs will happen in the testing phase, we have to change all phases.

Based on the defects of waterfall model problems we have another model called **Spiral model** which is also called **iterative model**. This is another type of SDLC model

**Spiral model or iterator model**:

**\*** Spiral model is iterative model

\* Spiral model overcome drawbacks of waterfall model.

**\*** We follow spiral model whenever there is dependency on the modules.

**\*** In every cycle new software will be released to the customer.

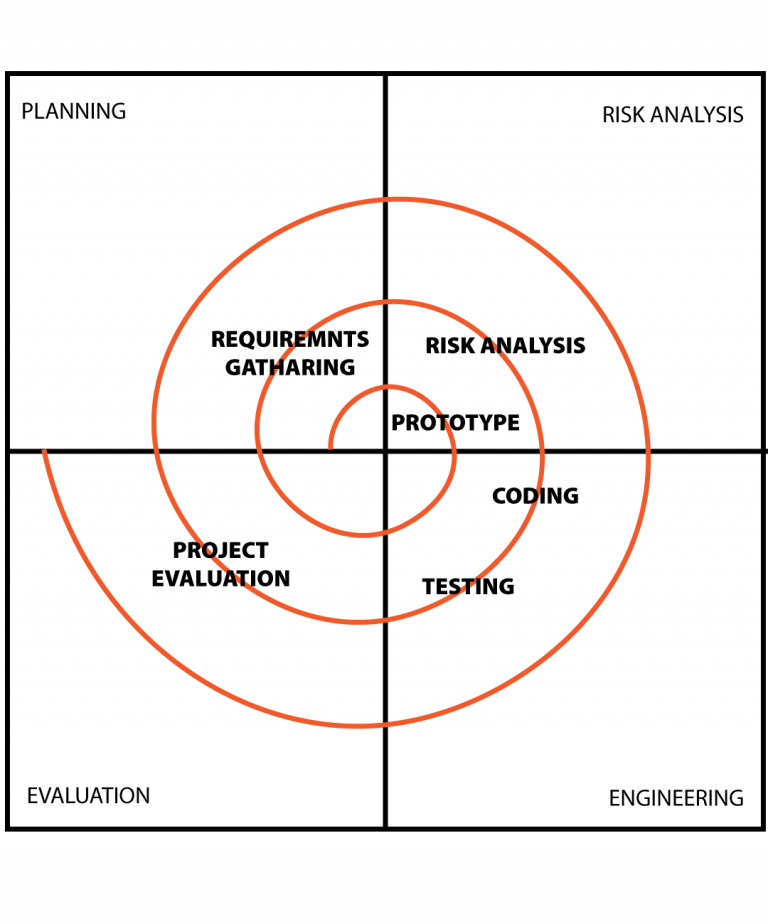
\* Software will be released in multiple versions. So, it is also called as version control model.

**Planning** - Here planning in the scene requirement analysis then after requirement analysis we design the code after completion of requirement analysis.

**Risk Analysis** - The risk analysis we will plan like how much risk is involved, how many developers and testers are required.

**Engineering and execution –** This particular area which is talking about design and development.

**Project Evaluation** – This particular area will be testing phase



How this spiral model works?

-> The process will be same on the SDLC, but the terminology is change, so, initial level we will estimate some cost and will do some planning this particular part this part is a planning means requirement analysis and understanding the risk analysis, budget cost everything we will thing about.

After completed development and testing will start and then release a software to the customer, customer evaluation.

So, once cycle is completed like this one cycle will having five phases once completed all 5 phases, we delivered a software to the customer and call it as first phase cycle.

Again, will top of this the customer will give a few more requirement then on top of the software again we do requirement analysis design development and testing in the second phase cycle and again we will release updated software to the customer.

Again, will implements the new requirements top of existing requirements again requirement design development and testing start, and we will release another version of the software in the next cycle that is a third phase cycle.

So, in every cycle we will follow the SDLC process, but the process will follow in the different cycle, so this is the behaviour of the spiral model.

We are able to accommodate the customer requirements some new requirements suppose the customer is giving initially 10 requirements and these 10 requirements we will complete one cycle an then we will deliver to the one software to customer, and he will keep on working those 10 features, first version is released.

Again, customer came back introduced 5 new requirements and second cycle we will implement the 5 new requirements on top of existing software this is the second cycle.

And similarly, even customer will be having some changes suppose he is comeback and introduce some new changes in the old requirement then we can also change those requirements in the second cycle.

So, the advantage of this model is the main intention of introducing of this model is to accommodate the new requirements and accommodate the frequently changing requirements.

Especially this model is preferred for **product-based company** because the product is a software which is developed based on the market requirement, so every time the market requirement will be change and accordingly, we have to implement the new features in the software also for this reason this model is best suitable for product-based companies.

For example, initial days what’s app is having very less or smaller number of features after that what happen every time will be getting some upgrade, upgrade means the new features so, based on the customer requirements will implement the new features and top of existing functionality or existing software they will release a new software to the customer. So, this is coming under iterator model. Iterator means the same number of phases repeating again and again with the new requirements.

What are the advantages and disadvantages of the spiral or iterator model:

**Advantages of spiral model:**

\* Testing is done in every cycle, before going to the next cycle.  
\* Customer will get to use the software for every module.

\* Requirement changes are allowed after every cycle before going to the next cycle.

**Disadvantages of spiral model:**\* Requirement changes are not allowed in between the cycle.  
\* Every cycle of spiral model looks like waterfall model.  
\* There is no testing in requirement and design phase.  
  
**Prototype model:** It is a blueprint of the software

**Initial requirements from the customer -> Prototype -> Customer -> design, development, testing ….**

In this model first we will get the initial requirements from the customer and after that we don’t develop the complete software we will develop prototype, once we develop the prototype we will show to the customer if the customer is satisfied with this prototype, then we will start design coding and testing etc.

So here the customer involvement is more because in every step the customer will give the inputs and based on that we will design the prototype and continue the rest of the parts.

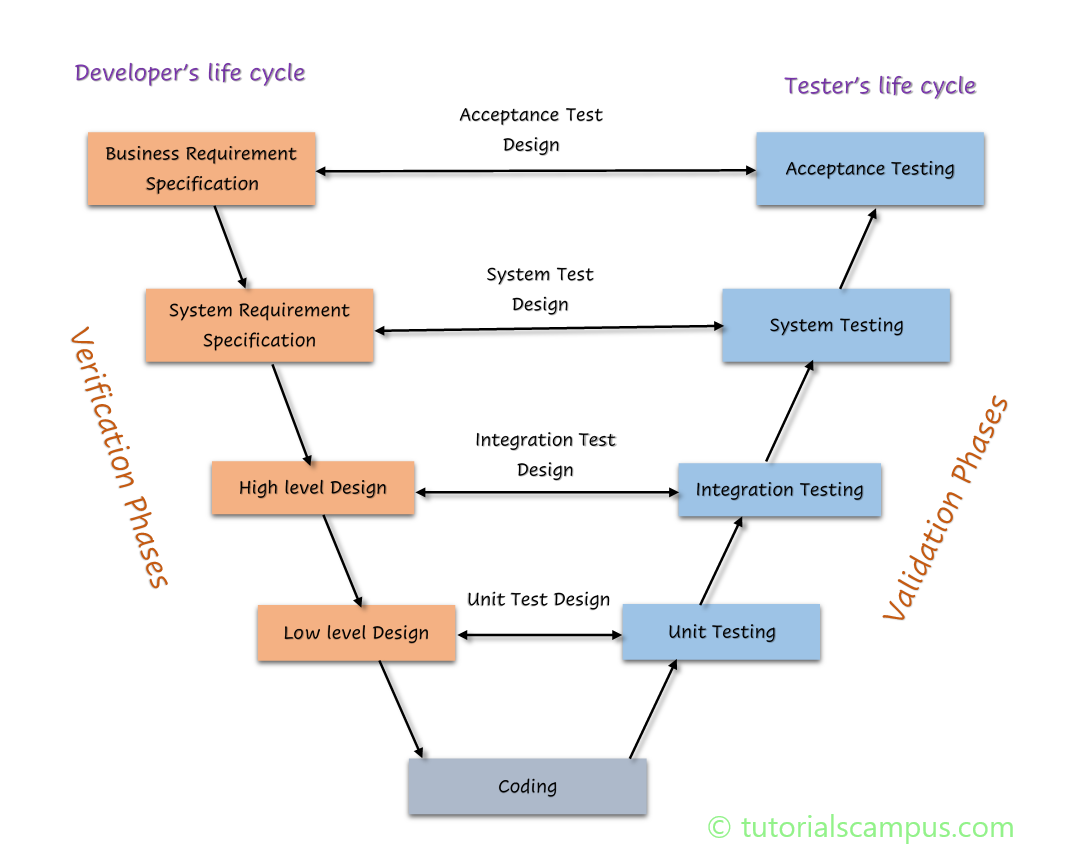
This model is coming between waterfall model and spiral model. Spiral is advanced than prototype model

It is also having some drawbacks which we can also overcome by using spiral model because if we have number of features if I want to accommodate in prototype, it is not possible.

What is module?  
-> Module is a part of the entire software will be divided into multiple parts. So, each part is called as module.  
EX: we are developing a banking application  
Login page, check balance, request statement, check loans, add payee etc. these are the modules we can make it a team for every module.

**V-Model or VV Model: (Verification and validate)**

In this particular model we can overcome all the drawbacks from waterfall model and spiral model.

So, why this model calls it as V-Model, because this model seems very similar to the V-shape.  


The main speciality of the V-Model is in every phase we will conduct the testing, in the requirement analysis phase we do testing in design phase we do testing, in development and testing phase also we do test so this is the speciality of the V-Model.

In above diagram one part is **verification** and the other part is **validation**

**Verification Part:**  
  
**First Phase:** We have a first phase in the software development process is collecting the requirement from the customer. Here we have BRS/CRS/URS (Business, Customer and User requirement specifications). The terminology is different, but the context is same.  
Basically, this document contains the business requirements. Whatever the requirements are given by the customer or client will be document in this phase. Business unit will be prepared those people interact with the customer or client collect the requirements and they will prepare the document this document more can be understand by the businesspeople not from the technical people.

So, based on this document we conduct one testing called User acceptance testing. The BRS/CRS/URS is the input for User Acceptance testing (UAT testing).  
Whoever is prepared BRS/CRS/URS this document and they are responsible for conducting the testing so the review the document and ensure the correctness and completeness.   
So, they review the whatever the content which is mentioned in the document correct or not and if it is correct then check complete or not all the requirements are mentioned in document or not that will be verified, that is a kind of testing.

First phase businesspeople (BU) will involve.

**Second Phase:** BRS/CRS/URS this document is not understand by the developers or testers, so we need a document which contains some more technical details, so based on this BRS/CRS/URS document there will be another document will be prepared that is SRS (Software requirement specification) document.

So, SRS document is a more technical details same requirement it contains the technical manner by using dataflow diagrams, Venn diagrams, Camunda workflow etc. Especially developers and testers will understand

Second phase (PM) Project manager involved; SRS document will prepare by product manager.

**Third Phase:** Based on the SRS document the designers will prepare HLD (High level design document) and LLD (Low level design document) document.  
HLD – This contains the entire application main modules.  
LLD – Low levels modules each main modules are divided into sub-modules of applications.

Third Phase Designers: HLD and LLD documents prepared by the designers.

By using **Review, walkthrough, and inspection** like we conduct testing on those documents, these three are testing techniques to verify the document.  
  
**Review:** Conducts on documents to ensure correctness and completeness. Review can be done anyone there is no restriction.  
\* Requirement review  
\* Design review  
\* Code review  
\* Test plan review  
\* Test cases review etc.  
  
**Walkthrough:** It is also kind of a review which is informal, means we don’t have a specific plan, meeting invite and whenever we want to do we can do walkthrough.  
\* It is an informal review.  
\* Author reads the document or code discuss with peers.  
\* It’s not pre-planned and can be done whenever required.  
\* Also walkthrough does not have minutes of the meet.

**Inspection:** Inspection is a more planned, means we have a pre-planned.  
\* It’s a more formal review type  
\* In which at least 3-8 people will sit in the meeting; 1 – reader, 2 – writer, 3 – moderator plus concerned.  
1 – reader: He is the author of the document. Whoever is created the document call it as reader  
2 – writer: Writer is nothing but suppose while reading the document by the author rest of the team will raise some questions and clarifications and the writer will note down all the questions and clarification what they have discuss in that particular inspection meeting.  
3 – moderator: It means he is like a mediator between the author and rest of the team. He is basically organizing the meeting.  
\* Inspection will have a proper schedule which will be intimated via email to the concerned developer/tester.  
Whatever we conduct on this document which is basically called as **static testing** techniques or testing the project related documents in the form of reviews, walkthrough, and inspection is called as **static testing**.

**Fourth Phase:** Here the development will start based on the documents. Once the development is stared slowly the software will be ready, so some modules will develop and after developing multiple modules they will integrate those modules and after integrating the modules they will conduct one more testing.

**Validation Part:**

**Unit testing:** Whatever the module which is developed by the developer if we test that particular module which is called as a unit testing.  
Unit testing in the sense testing the single module or single component of our software which is called as a unit testing.  
So, developer himself to conduct this unit testing. Because they know the coding part, they know how to write the code so, they know how the module is working.

**Integration testing:** After that integrate all modules and make the one module, then will test that module which is called as integration testing.   
These two Unit testing and Integration testing will test developer itself. For this two testing we have to know the coding part, programming language.

White box testing – Unit testing and Integration testing will be conducting at the code level which is called as a **white box testing**. It means we know what is inside the box

After completion of integration testing the final integrated software part will be ready and that software will be given to the testing team then system testing will be conducted by the testers.

**System testing**: Here the testers will be involved to test the software. But what type of testing they do basically they don’t need to understand the code, they will just verify and test our software is working into customer requirement or not, so what customer accepted what are flows we have, what is the functionality weather is working or not that will be tested in the system testing phase.  
The actual testers will be involved in the system testing phase.

Black box testing – We don’t need a programming language, here we are getting a complete application we don’t know exactly what developers written internally, don’t know the code that is called as **Black box testing**.  
It means we don’t know what is inside the box.

**UAT (User Acceptance testing):** After completion of testing, the user or customer will do some kind of testing. This also done by the testers along with customers, here also we don’t need to understand the code which is also comes under black box testing.

After developing the software, we are doing unit testing, integration testing, system testing and user acceptance testing, so these testing are called as **Dynamic testing**. It is where we actual testing the software.

There are four kinds of techniques we have are:  
**Unit testing, Integration testing, System testing, User accessible testing**

**Unit testing:** During unit testing developer will concentrate on testing the code what they have written.  
  
**Integration testing:** During integrating testing developer will integrate the modules and multiple components in the software and they will check the data flow between the components weather those components properly communicated or not this is integration testing.  
  
**System testing:** Actual testers will perform the system testing weather the software is working according to customer requirement or not, meeting all the requirements or not, the functionality is working perfectly or not, performance is good or not, security testing, UI testing lot of testing done by the testers.

**User accessible testing** (UAT): Normally UAT will do by the testers along with the customers and they will set-up the exact environment where the customer is going to work and install the software and from the customer point, they will test some flows like what they do day to day basis on the software, what are the transaction they will test them, and also executes some of the test cases on the software this comes under the UAT testing.

**What is the difference of Verification and Validation?**

**Verification:  
\*** Verification is checks weather we are building the right product.  
\* Focus on the documentation.  
\* Verification is typically involved:  
 \* Reviews  
 \* Walkthrough  
 \* Inspections  
\* Static techniques are involved in verification testing.

**Validation:**\* Validation is checks weather we are building the product right or not.  
\* Takes place after verifications are completed.  
\* Focus on the software.  
\* Validation typically involved  
 \* Unit testing  
 \* Integration testing  
 \* System testing  
 \* User acceptance testing  
\* Dynamic techniques are involved in validation testing.

**What are the Advantages and disadvantages of V-Model**

**Advantages:** Testing is involved in each and every phase **Disadvantages:** Documentation is more, and initial investment is more

**What exactly QA (Quality assurance), QC (Quality control) and QE (Quality engineering)?**1) Difference between QA and QC  
 \* **QA** is process related (QA - Process)  
 -> 1) It means there is a company 3 important pillars are People, Process, and Product (3P’s).  
 The QA is belongs to the process related and they will define the process, which means high level  
 management people is comes under the QA, which is talking about entire development process  
 from the beginning, requirement to till reach we release to the product for customer, so the entire   
 process will designed these people. And also, these people are making sure rest of the people are   
 following the process properly or not.

2) QA focusses on building in quality.  
 3) QA is preventing the defects.  
 If we follow the process correctly or perfectly, we can prevent the defects in the future that is a   
 prevention activity.  
 4) QA is a process oriented.  
 5) QA is entire life cycle.

\* **QC** is the actual testing of the software. (QC - People Testers)  
 -> 1) They are involved only during testing part, people who are working under testing actually comes   
 under QC. Those people actually test the software weather it is working or not according to customer  
 requirement.  
 \* QC actually talking about the people.  
 2) QC focuses on testing for quality.  
 3) QC is detecting the defects.  
 It means the bugs will be there we are trying to find out that as detection.  
 4) QC is product oriented.  
 5) QC for testing part in SDLC.

2) **QE (quality engineering):** This term is recently started.  
 -> This is little advanced term than QC, after QC QE will come.  
 People are belonging to QC like testers they do normally manual testing and they also do some   
 automation testing, testers belong to manual and automation testing these are belongs to QC.  
 -> QE means whoever is doing automation testing normally they will write a code but testing the software   
 those people are comes under quality engineering.  
**Ex: Software engineer**/ SE and **Quality engineer**/ QE  
When the **software developer** what they have do normally they will write the code using programming language to develop the software, the main focus on development.  
What **quality engineer** will do also write the code for testing the software, QE we can call it as a automation testers.  
QE is mostly related to QC itself.

**What are the levels of testing?**  
-> There are 4 levels of testing are:  
1) Unit testing  
2) Integration testing  
3) System testing  
4) User acceptance testing

1. **Unit testing:**  
   \* A unit is a single component or module of a software.  
   \* Unit testing conducts on a single program or single module.

\* Unit testing is a white box testing technique.  
\* Unit testing is conduct by the developers.  
\* Some of unit testing techniques are:  
 \* Basis path testing: Suppose developers created some program which is having multiple lines and   
 when it’s testing this program each and every line should be executed at least once, so that comes   
 under the basis path testing.  
 \* Control structure testing:Normally in programming we have If else condition, switch case condition so we will verify those   
 conditions are working properly in all the cases we pass negative and positive conditions.  
 \* **Conditional coverage**: The developer is verify the condition in positive input and negative input  
 Ex: a = 20, b = 10  
 if a>b  
 print a is largest  
 else  
 print b is largest  
 \* **Loops coverage**: Sometimes in the programming if we want to repeat some statement we have to   
 keep in the looping loop block and that statement will be repeating multiple times.  
 Ex: I want to print 1-5 numbers I = 1 max = 5   
 while(I<=5) {  
 print(I); I = I+1  
 }//1 2 3 4 5   
 Now, in the loops coverage when I execute this program weather it is starting from beginning or   
 not and it is repeating exactly till the max number or not and it printing all the number with range   
 or not we are going to verify.  
\* Mutation testing: Mutation is nothing but a repetition, suppose some program we want to test we want to provide different type of inputs.  
Ex: if user = “Scott” and password = 123  
 allow access to login  
 else  
 Not allow login   
So, in mutation testing we will testing this code to passing different inputs

**2) Integration testing:**   
 \* Integration testing perform 2 or more modules  
 \* Integration testing focuses on checking data communication between multiple modules.  
 \* Integrated testing is a white box testing technique.  
 There are two types of Integration testing are:  
 1) Incremental Integration testing  
 2) Non – Incremental integration testing  
 **1)** **Incremental Integration testing:** Most of the times we are used Incremental integration testing.  
 It means incremental adding the modules and testing the data flow between the modules.  
 There are two approach of Incremental Integration testing are:  
 \* Top down and \* bottom up

**Top-down Incremental integration testing:** Incrementally added the modules and testing the   
 dataflow between the modules. And ensure the module added is the child of previous module**.**

**Bottom**-**up Incremental integration testing:** Incrementally added the modules and testing the   
 dataflow between the modules. And ensure the module added is the parent of previous  
 module**.**

**Sandwich/Hybrid Approach:** Combination of Top-down and bottom-down approach is called  
 sandwich/hybrid approach.

**2) Non – Incremental integration testing:** Adding allthe modules in a single shot and test the data   
 flow between the modules.  
 **Drawbacks are:  
 \*** We might miss the data flow between some of the modules.  
 \* If we find any defect, we can’t understand the root cause of defect.

**3) System testing:** This is the actual area testers will be involved  
 \* Testing over all the functionality of the application with respective client requirements.  
 \* It is a black box testing technique.  
 \* This testing is conducted by testing team.  
 \* After completion of component and integration level of testing’s we start system testing.  
 \* Before conducting the system testing we should know the customer requirements.  
 \* System testing focuses on below aspects:  
 \* User Interface testing: It means UI of the application, we going to test on the UI part all images  
 properly aligned or not, logo is displayed or not, application elements like checkbox, drobox,

Dropdowns are properly displayed or not, look and feel of our application colours and fonts etc.  
 these things we are mainly focusing that is UI testing.

\* Function testing: Here we are going to test the functionality of the application.  
 Functionality means the navigation suppose the banking application login is the one functionality   
 after login we will test the balance module functionality, and money transfer, adding payee,   
 generating report, request check book functionality same like lot functionality is there we are test  
 those functionality is properly working or not.

\* Non – functionality testing: In this sense we are testing about security, performance, installation   
 compatibility testing, etc. so these are all comes under non-functionality testing.  
 Once the functionality testing is stable then we are doing non-functionality testing.  
 This not done by the normal testers, they are separate dedicated team will be there to conduct  
 non-functionality testing, this needs a different set-up environment.

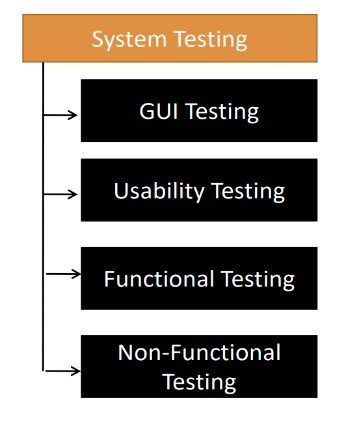
\* Usability testing: In this sense user help, suppose if we bought some product like phone, watch tv  
 etc. Along with those products we will also get some user manual documents. It contains how we  
 use a product this document is very important to required to the users. So same way along   
 with the software we are also going to provide the user manual to users or customers.  
 We have to verify those user manuals, like proper content or not, how to install the product how  
 to perform the functionality each and every step whatever we do in the application will be   
 documented that is basically called as a **user context**.  
 Some times in our applications also we will see some menu at the corner even if we open the  
 notepad whatever we can see help menu there can get the help of the particular product.  
 So, these are the things we are going to test the usability testing.

**4) User acceptance testing:** After completion of system testing user acceptance testing will be conduct  
 It will normally conduct by the customer or users who are using the software and everything is   
 working file and they will accept the software, whatever the requirement they have given finally   
 verify those requirements are satisfied with this product or not, so the main testing is done by the  
 users or customers or testers and developers can be do some help on them.  
 Sometimes testers also involved along with the users when the user acceptance testing.  
 There are two types of testing are do UAT team are:  
 \* Alpha testing \* Beta testing

Alpha testing:In this sense the users or customer do the testing on dev or testing environment.  
 means what they will come back to the company wherever the software developed and do some   
 testing that is alpha testing.

Beta testing: After that they will get the software and install that in customer environment and   
 they do some basic testing that comes under the beta testing.

After completion of these two testing our product or software is go to production environment.  
 and then the actual customer will be using the software.

**System Testing:**

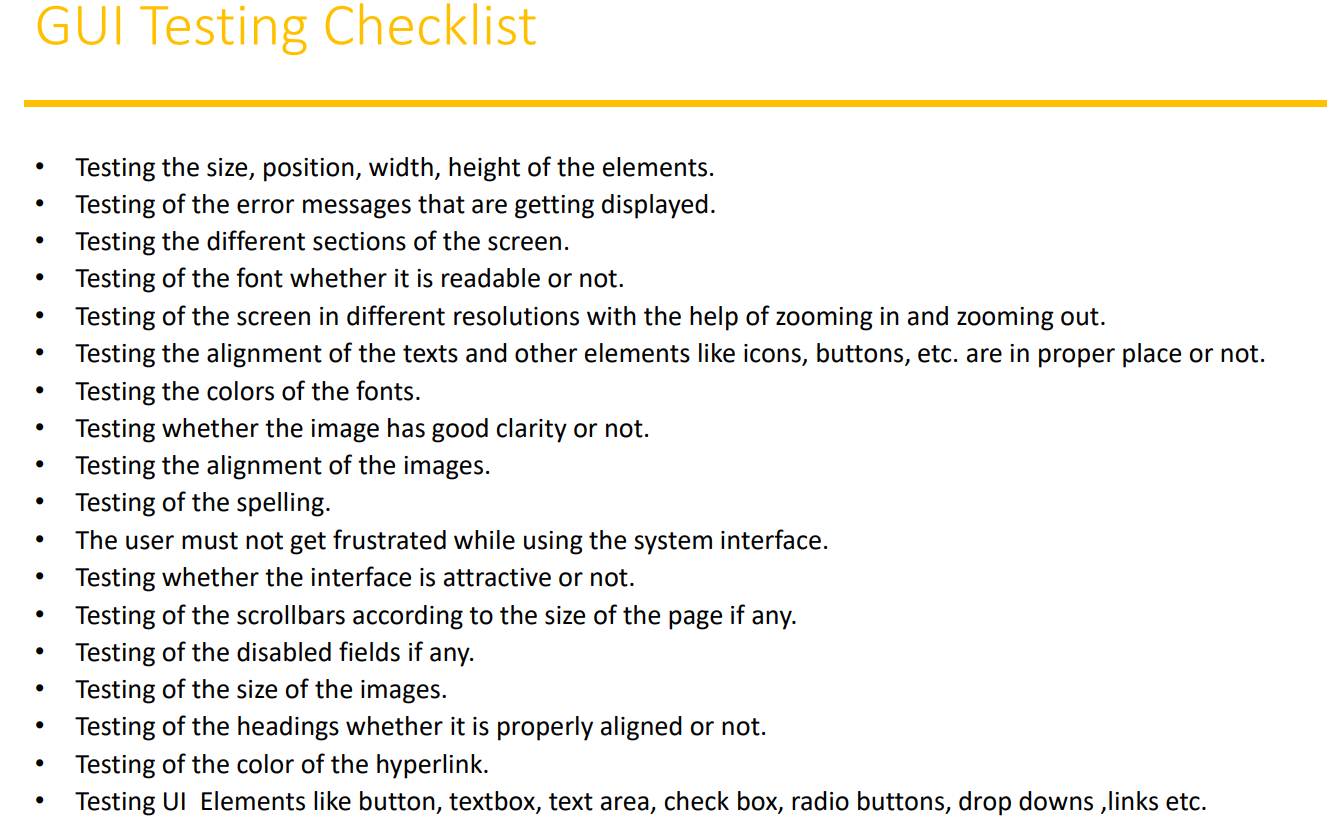
Basically, system testing means we are going to test the overall functionality of the application with respect to customer requirement.  
In system testing we are mainly focusing of 4 different areas are

**1) GUI testing (Graphical user interface)  
2) Usability testing  
3) Functional testing  
4) Non-functional testing.**

In the above four system testing, we are mainly focusing on **functional testing**.

1. **What is GUI testing (Graphical user interface):**

\* GUI testing is a process of **testing** the user interface of an application.  
\* A graphical user interface includes all the elements such as menus, checkbox, buttons, colours, fonts, sizes, icons, content, and images.



1. **Usability testing:**   
   \* During this testing validates application provided context sensitive help or not to the user.  
   \* Checks how easily to the end users are able to understand and operate the application is called **usability testing**.

It is also called as **context sensitive help** It means wherever we stocked in the application immediately we get the help. So, every application or software must provide help to the customer. Even If we buy some products, we get the user manual because it contains the functionality of the product.

Each and every step will each and every thing should be mentioned in the user document, for the user point of they can understand the document it should be very simple English and everybody even no technical people also able to understand and reading the document.

They should work on the application while understanding the document without depending on any other people.

For creating this user manual document developer involvement is main important.

1. **Functional testing:**

\* Functionality is nothing but behavioural of the application.  
 It means weather its working according to customer requirement or not that is functionality, we are going to test those functionalities. Our focus is not on the UI elements our focus is functionality of the elements.

According to the we prepare from test data, and we are verifying the test box is allowing this type of data into the requirement, so that is the functionality.

\* Functional testing talks about how your feature should work according to the customer requirement.

\* As part of the functional testing where exactly focus on our application:  
 1) Object properties testing:

Every element is on the web page is having certain number of properties like **input text box** is   
 enabled/disabled **radio button** is selecting the any one or not, **dropdown** button weather it is

Selecting one element or not that we have to verify. Sometimes we have a **combo box** in that   
 we are selecting more than one options weather its selecting or not and observe suppose login   
 page once we entered a username automatically our cursor goes to the second text box and  
 waiting for the password this called as a focus attribute or focus property. So, these kind of things   
 we have to verify.

2) Database testing/ Backend testing:  
 It is another major area we have to test. Here we need know some basic SQL command   
 knowledge.

It’s a huge kind of testing it is not a simple testing, here as a function tester they can do some   
 basic database testing like functionality level, but internally we have done lot of testing, so we  
 have separate or different skillset is required to perform the database testing.

There are lot of things are involved here, but as a tester initially we mainly focusing on the **DML   
 operation (Data manipulate language)** As a tester in UI side performing a **CRUD** operation or   
 **insert, update, delete, select** when do these operations from the UI, those operation is properly  
 working at the database level or not that we are going to verify.

After we login to the database and execute certain number of queries to retrieve or to see the data   
 from the table. The UI we do the black box testing an in backend we are login to the DB and   
 execute a query to check that data and this comes under the white box testing area.

So, the database testing includes white box and black box combinedly we can say   
 **Grey box testing technique.** Whenever we do the combination of white box and black box testing, we call it as a   
 grey box testing.

Apart from this we are doing lot of testing in database are table level validation, how many   
 columns we have in the table and what kind of data we can insert in the table, size of the column  
 number of rows are available, number of fields are available, sometimes multiple table we have in  
 database data we are submitting the UI that can be stored in the multiple table, even that   
 relation of two tables we verify, and also verify some, functions, procedures, triggers there are   
 lot of other concepts we have in the database all these thing we have to verify.

What exactly in database testing is:  
 \* Table level validations (Column type, column length, number of columns….)  
 \* Relation between the tables (Normalization).  
 \* Functions, Procedures, Triggers, Indexes, Views etc. these are the important things we have to   
 focus while performing the in detailed database testing.   
 These are coming under the white box testing.

3) Error Handling: **\*** It means where we have to mainly focusing on error messages.  
 \* Testers verify the error messages while performing the incorrect action on the application.  
 \* But the error messages should be readable, understandable and should be in the simple   
 language that is also we have to check.  
 **Ex**: working something on UI when we do some invalid action or when we do some incorrect actionor incorrect operation on the UI our application displayed some message to users.  
 Error message should be displayed on UI, otherwise user don’t know exactly what is went wrong,   
 what is the mistake done. So, error message also most important while testing the application.

4) Calculations/Manipulations testing:

**Agile Testing process and Terminology:**

\* It is an iterative and incremental approach  
\* iterative means same process repeating again and again  
\* Incremental means, modules/features keep on adding on top of existing software  
\* There will be a good communication between the customer, business analyst, dev team and QA team  
\* The goal of agile model is the customer satisfaction by delivering the piece of software to the customer within short span of time.

**Advantages:**

\* Requirement changes are allowed at any stage of development  
\* Release will be very fast (weekly)  
\* Customer no need to wait for long time  
\* Good communication between team  
\* It is very easy model to adopt

**Disadvantages:**

\* Less focus on design and documentation since we deliver the software very faster

In Agile we have some frameworks, we are following as a SCRUM framework

**SCRUM:**

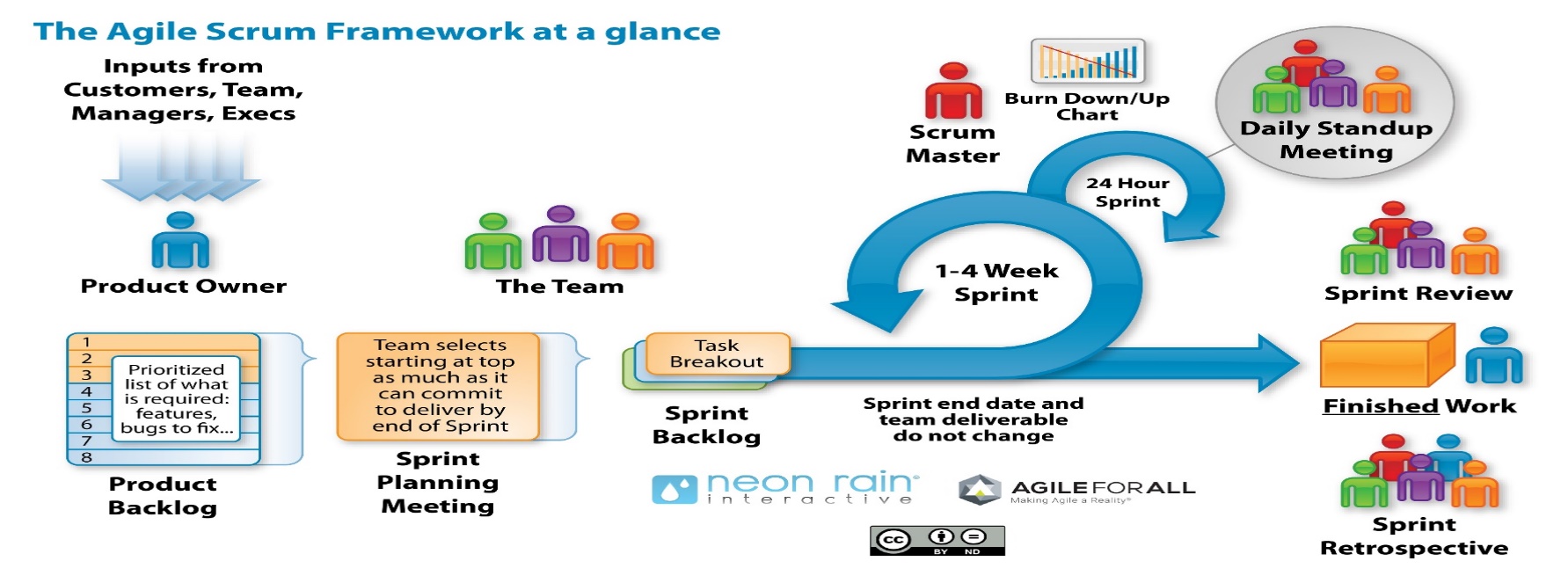
Scrum is a framework through which we build software product by following agile principles.  
Scrum includes group of people called as scrum team.  
\* Product owner  
\* Scrum master  
\* Dev team  
\* QA team

1. **Product owner:**

\* Define the features of the product  
\* Prioritize features according to market values  
\* Adjust features and priority every iteration, as needed  
\* Accept or rejects work results

2) **Scrum master:**   
 \* The main role is facilitating and driving the agile process

3) **Developers and testers:** \* Develop and test the software



**Step – 1:**Product owner is capturing all the feature or functionalities from the customers or stakeholders, and he will prepare a small document called **product backlog document.**

Product backlog is just as an excel sheet it is not having big kind of document it will be small and short form document.

In product backlog contains requirements and features which we need develop and test, but in agile process we don’t tell requirements and features etc, we call them as **epics and stories.  
Stories** is nothing but a small requirement (Just like a unit)

**Epics** is nothing but a big requirement. Means epics is a collection of stories. (Just like a module)

**Step – 2:**Sprint or iterative planning meeting, in this meeting product owner will be part of this meeting developers, testers and scrum master all the people will involve in this meeting.

In this meeting the QA and dev team sit together do a **story estimation** based on that picks the stories to product backlog whatever the stories they want to develop and test within 2 – 4 weeks of time and they prepare **sprint backlog** document

**sprint backlog:** It contains a stories collected or selected by dev and QA team for that particular sprint. Or   
 sprint backlogis a subset of product backlog.

Sprint backlog will keep changing from one story to another, but product backlog will not change once it is created and approved by product owner that will not change.

**Step – 3:**During the sprint cycle developers will develop the stories parallelly testers will do the testing activities both dev and QA activities will not change, during the spring cycle every day we will have 15 min of meeting will all the team this is called **scrum meeting**

In the daily scrum meeting we will discuss three things are:

\* What I have done yesterday   
\* What is the plan for today  
\* What we are going to do tomorrow

Or if in case any issues, any blockers those things we have to discuss during the daily scrum meeting.

Scrum master will conduct this meeting every day we need to report all the status, progress everything to the scrum master during the meeting.

so based on the spring planning within 2 – 4 weeks of time the development and testing should be completed, we must finish the work, once finished the work two things will happen are:

1. **Sprint review meeting**  
   During the sprint review what we need to do is we have to give the **demo to the product owner** because product owner is the guy reject or accept the work.  
   \* During sprint review meeting like what are stories we are planned to initially weather done or not and if done we provide a demo for product owner.

\* In the demo product owner satisfied or not.

\* How many bugs we are found during the sprint and what is the impact of those bugs like high priority, low priority what type of bugs are there, and we fix these bugs again we plan next sprint to fix those bugs.

\* Once the demo is done product owner is satisfied with that, then we release small piece of software to the customer. Then customer start using the particular software. With that one sprint cycle will be completed.

1. **Sprint retrospective meeting**

In this meeting, after customer is use the software then we take feedback for them based on that  
we discuss about mainly three things during this spring planning:

**\*** What went well

**\*** What went wrong

\* Improvement areas

With that we consider as completed as one cycle

In the next sprint again from the product backlog some more stories will be captured, again one more sprint planning will be conducted in the next cycle, again 2 – 4 weeks and if there any bugs open in the previous sprint do estimate from those bugs and every day meeting will be conducted by the scrum master after completion of sprint again we have spring review and sprint retrospective meeting after that we deliver to build to the customer.

So, in every cycle we will implements some piece of stories and will provide the build or provide a piece of software to the customer.

In the daily scrum meeting the Scrum master will prepare some charts burndown/burnup charts like how much work we are planned, how much work we have completed, so that is representation the work will be calculated and representing the form of charts.

Based on that charts the scrum master know exactly, how the process is going on or not.