Teacher: Mr. Kuhzat

**What is a software?**

The applications like Amazon, Bank of America and others are software.

All software are mainly used to solve problems. Every software is solving problems and making the work easy.

* How do you solve a problem?
* How much do I need?
* How long it will take?

**Example: Receipts**

1): I have to keep too much Receipts at home, we have to have an application and resolve this problem and for that reason we need the following steps:

* [www.eReceipt.com](http://www.eReceipt.com)
* Requirement
* Technical
* Design
* To solve the problems

Client and consultant or vender:

* Requirement Engineering
* Business Analyst – what is in client mind and what they can develop. They write it down and in a specific format.
* Developers, put them in a computer language and develop it as a software application
* Developers speaks software languages

What is Requirement?

* Functionality
* Expectations of users, describing of expectation is a requirement.
* The process to gather the software requirements from client, analyze and document them

Why feasibility study?

* It means if the project is working or not.
* Understand whether the project (product or service) is actually in demand.
* Assessing whether the proposed product or service is marketable
* What is the return of investment?
* Estimate the project’s likelihood for success

Requirement Gathering:

* Where to gather requirement?
* Talk to the customers
* Instagram example?
* Talk to End-Users
* Talk to Domain Experts (a close application which may function like your application for example Uber &Lyft
* In order to make a good application, prepare good requirement.

Smart Criteria:

(S): Specific:

* The more specific requirement you have, it will make it easy for the customers.

(M): Measurable:

* If you have more users, it takes more time, you have to write it very clear. For instance, the website of immigration of Canada was crushed because of the millions of users using it simultaneously. The web developer didn’t specify the number of users and it is very important.
* Famanie lost 200 million dollars because they put a loan number only 6 digits, once it arrives at 999,999. The 300,000 banks in USA couldn’t ask for money and they lost a lot of money.
* When there is a problem for an application, they call it hot fix like; Amazon or banks.

Requirements:

* Functional
* Non-functional

Requirement Characteristics:

* Clear
* Correct
* Consistent
* Unambiguous
* Modifiable

When the testing should start?

* As soon as we have the requirement documents and it starts from the idea stage.
* The requirement must be very clear.
* Verify the requirement. You will have better products based on the better requirements if you write very specific requirement.

Verification and validation:

* It is very important part in SDLC.

Evaluation Items:

* Plans
* Requirement Specifics
* Design Specifics
* Code
* Test Cases
* The actual product or software

Activities:

* Reviews
* Walkthroughs
* Inspections
* Software developers explain the projects and the business analysts stop them and ask questions about the details of functionalities.
* Peer Review: To check other codes or products and your products is checked or evaluated by the colleagues and classmates.

SDLC (Software Development Life Cycle):

* Idea
* Gathering & Analyzing Requirement
* Verification and validation requirements
* Smart Criteria
* Design (the architecture software engineers do design, they have big white papers, they draw everything like designing a building and then they make the application to function). The design phase comes after a good understanding of customer’s requirements
* Coding (which puts all pieces to gather to perform specific functions).
* Testing
* Software in production & maintenance
* To build a specific functionality or deploy a code from a development stage, the developers test 80% testing before the code is deployed. They prevent the deficiencies prior to release.
* If you learn Java very well to be to perform the Unit testing properly. It makes the developers job easy and you can be hired quickly by the companies.
* During the interview tell the developers that you can test clean codes.

**Testing:**

* Write Test Cases
* Execute Test Cases
* If the Actual result didn’t match the expected result, then there is a Bug in System.
* Our job is to write test cases
* Log Defect(Bug): (take screen shot and report to logging defect).
* Retest existing defect, if has the same problem.
* Automate Test Case
* Execute Automated Test Cases
* Verify if the software meets customer expectations
* UET: User’s Excepting Test

Personal Value Preposition (PVP):

* You have to sell yourself and explain your capabilities for the companies in order to hired in higher capacity.

How do you execute the test cases?

* There is a triangle green button, you just click run
* We use from Eclipse, there are written codes in Eclipse which helps make your work easier during the actual job.
* Once testing is done, then the Software will go to Production.

Terms that are used mainly during the interview:

* It is better to use the word “It Depends” during the interviews. For instance you say it depends on the project, in our current project we do xyz and in other projects we do yzx.
* Instead of saying I read, you say I analyze
* Instead of saying I write codes, you say I develop the software language.
* In interview they ask you to talk about yourself for 60 seconds, you must use the strongest & the most professional words to just sell yourself and be hired. The interviewers should take a positive feeling from the initial introduction.

Environments:

1. Dev Environments ([www.dev-amazon.com](http://www.dev-amazon.com))
2. Test/QA Environments (deployment to the production)
   1. [www.qa-amazon.com](http://www.qa-amazon.com)
   2. [www.test-hc](http://www.test-hc)
3. Production environment

Teacher: Mr.Irfan

Email address: Irfan@cybertekschool.com

**Testing**:

* Why do we do testing? Before we do something, first ask yourself why you do what you do?
* What, how and why? These apply to every person’s life in all aspects.

What is Quality:

* Quality is a value judgment assigned to an object, a place or an idea based on it’s worth to the observer.
  + Quality is Perception
  + It is very important to know the perception of the customers.

Software Testing:

* Software Testing is a process of executing a program or application with the intent of finding the software Bugs (ISTQB).
* Process of validating and verifying that a software program or application or product is functioning fine.
* Meets business and technical requirements that guided its design and development
* Maybe the requirement document is not clear, because the human being is making the requirement and you have to work on it.
* Works as expected
* Make sure 100% of customer and end-user satisfaction
* Software testing is a process and activity to make sure the product we have developed will meet customers and end-user’s expectations

**Testing Types**

Static Testing:

* It can test and find defects without executing code. Static testing is done during verification process. This testing includes reviewing of the documents (including source code) and static analysis. This is useful and cost-effective way of testing. For example: reviewing walkthrough, inspection, etc.
* The testing starts from the very beginning stage.

Dynamic Testing:

* In dynamic testing; you actually do the testing, write code or whatever is required to run the whole test.

What is a Bug:

* Bug; is like a mistake, error or malfunction in software field. Website crashing, or you search something and something else pops up.
* Bug; is an expected result.
* The software doesn’t do something that the product specification says it should do. If you put a blue button to do certain function and ends up doing something else.
* The software does something that the product specification says it should not do.

Example:

* Amazon Prime is crashed, and during search the dogs show up and people don’t like it. The Amazon is handling the problem now.

Book:

* Software Testing by – Ron Patton

Why Bug Occur?

* Specification is not good
* Design is not good
* Coding is not good
* Others; maybe something else.

The cost of the Bug:

* If you find the bug at the beginning, it could be very cheap, if you find the Bug later on during other stages like; Requirement Specification Stage, Design Stage, Code, Testing or during Production Release Stage, then it will be that much expensive. If the issue is found at the early stage, the cheaper it could be, as it goes on, then there will be more people involved and it will be expensive.

What testers do?

* To find bugs
* To confirm things working fine
* Test to pass, you test something to pass it.
* You are the final validator, you are testing to validate the component.
* Tester should have test to break approach instead of test to pass approach.
* Testers should think out of the box.
* The goal of testers are to find Bugs.
* You must find the bugs in early stages, big corporations want to remove Bugs, not to lose money.
* Make sure to fix the Bugs as soon as possible.
* Once you identify the Bug, report to the Developers and keep watching and working with the Developers until it is fixed.
* Good testers should not afraid developers and business analyst, they should able to backup their opinion to make sure deliver top quality software. You can go to end-user if the developer and business analyst don’t listen to you.
* Always have documentation to be traceable to cover yourself. Example; send an email to the whole group explaining the Bug.

Risk Based Testing:

* The earlier you test a component, the low risk you can have.

White box testing:

* White box testing is the software testing method in which internal structure is being known to tester who is going to test the software. If you exactly know what is going on behind, then it is a white box testing. Examples: Testing smartphone chips, memory etc.

Black Box Testing:

* The tester only knows what the software supposes to do, he/she doesn’t care what is going on inside the box. No need to know the logic. Only make sure the software does its job. Example: Testing Smartphone by using it. Can I make a call using smartphone?
* Purple Box Testing:

**Testing Mythologies:**

* Functional Testing: it is the people who are doing test manually, Example, can user do this or that?
* Non-functional
* Performance testing: is testing that is performed, to determine how fast some aspect of a system performs under a particulate workload.

1. Unit Testing:

* A unit is the smallest testable part of an application like functions, classes, procedures, interference. The developers usually do the testing, the developers do Unit testing before they give it to Testers. Testers usually learn the Unit Testing. You can Unit test your own Automation.
* You don’t really have to worry, but you need to know what is Unit Testing?

1. Component Testing:

* Component Testing is a Method where testing of each component in an application is done separately. Amazon; Search, List, Add to cart, all these are components.
* If something is broken in component level, then we do not move on.
* Component testing is also known as module and program testing. It finds the defects in the module and verifies the functioning of software.
* Before integrating the components of the software, the single component has to be working fine.

1. Integration Testing:

* Test after integrating multiple component if the system works fine or not.
* Sometimes, maybe component and unit work fine, but when we run system it may not work. Example: updating of phones, any company once develop an application, it is not perfect, it is being updated as time pass.
* Integration Testing is usually larger than the Component and Unit testing.

1. Acceptance Testing (UAT User’s Acceptance Test):

* After the system tests has corrected all or most defects, the system will be delivered to the user or customer for acceptance testing.
* Deadline in IT is very important. You have to do it on time.
* Usually done by users or customers although other stakeholders may be involved as well.
* Also known as UAT User Acceptance Test

1. Smoke Test:

* Can be called multiple names;
* Smoke Testing Is initial testing process exercised to check whether the software under test is ready/stable for further testing.
* Before putting the entire team to testing effort, the system needs to be stable enough or worth to test. (the automation should check everything before the users use it).
* Automation sends an email to all testers that the smoke test is done. You make sure that the application is working before starting your job. You do it every morning.
* Smoke Test is to check the basics of an application.
* Smoke Test is also called Sanity Check.
* What kind of smoke test you do? I check the fundamentals of my application.
* Smoke Test is usually takes 15 minutes, if it runs for 2 hours, then there might be a problem (Bugs, Defect, error etc.).
* In Fannie Mae the smoke test is called Shakeout. So every company could have a different name for the smoke test.
* Perfect candidate for test automation because it will quickly scan the application many times every day.
* The developers do the unit tests, he makes sure it works, then he puts it in big application, the tester do smoke test before running the component testing to make sure the application is working.
* Developers (Dev) & Ops team can kick-off the smoke test to check if the build is stable.
* The codes will not work unless you build it or compile in Java. Build is when the developers deploy a function or application.
* Smoke Test can be scheduled multiple times a day to give timely report to the team.
* It can be triggered by Continues Integration Server.
* It can be triggered by sending email in outlook.
* The smoke test usually takes 15 to 30 minutes, if it is more than that then there might be some issues.

1. Ad-Hoc Testing:

* Also called monkey testing or Random testing.
* Ad-Hoc testing is testing without any plan or documentation.
* Is considered as informal testing
* It can be also called Exploratory testing. You explore the application to find bugs and other problems.
* Good amount of defect can be found in ad-hoc testing. Example; full screen, half screen and quarter screen testing of application.

1. Regression:

* Is from A-Z, every possible scenario about your application.
* New feature, Modification, Update, Bug Fix, Anything Else
* Before we do release, we do regression test, before it goes to the public the regression is done. It is like full application test.
* It is the last step of testers job to do regression test before the application is released for the public. Usually, every 3 months the release happen. The release happens during the night when the users are not using. If the application goes down for 20 minutes, the big application can cause loss of a lot of money. Example; Bank of America.
  + Regression, if any functionality is added we have to make sure the new functionality did not break existing functionality.
  + If BUG is fixed, then we have to make sure the bug did not break other functionality that was working fine. You have to run a regression test to make sure the whole system is working.
  + If there is any change in the code, then we have to make sure that developer did not break the other codes unintentionally.
  + Scenarios are just each step.
  + Regression testing suite will have a lot of test cases. We have 2000 test cases in our current project.
  + Regression is repetitive and very time consuming. It could take 3 days, it could take a week.
  + Regression test cases are perfect for test automation. To save time and if we save time we save money.

1. Security Testing: It is more from security point of view, like make sure the hackers are not hacking the system.

Teacher: Mr. Shaun

[shaunp@cybertekschool.com](mailto:shaunp@cybertekschool.com)

**Manual Testing and Basics:**

* Test Automation Training
* Cross fractional – someone who works multiple jobs in the same time.
* QA: Quality Assurance; you verify the product and assure that the product is good to go.

Smoke Test:

* Why smoke test is so important?

Example: Amazon example:

* Typing [www.Amazon.com](http://www.Amazon.com)
* Then check the other details.
* The smoke tests results are brought to the Automation to fix them.
* Virtual machine; it is a machine which is somewhere and all employees have access to it. Like; cloud. Prior to Cloud “Virtual Machines” were in use.

Ad-Hoc Testing:

* Random tests or functions that are asked by employees on daily basis.
* Sometimes the developers analyze a code and they want to make sure that the code is working.

Regression Test:

* The final test, after modifying of each component that the rest of system is working like before.

System test:

* It happens before the release, it checks all features from A to Z to make sure everything is working fine.

Software Testing Hierarchy:

* Unit Testing: Performed by developers … also called black box testing.
* Component Testing: Testing each component or module to make sure they are working as expected.

System testing:

* + Smoke Testing (Shakeout)
  + Functional testing
  + Regression testing
  + Positive testing
  + Positive testing: you are making sure that everything works as expected. You put a correct user and password and it works fine.
  + Negative Testing: if you log in with an in-correct user and password then it is negative testing.
  + Performance Testing: The number of using the application can use in the same time, how much load can the application take in the same time. Like; ObamaCare website which failed.
  + Many other types of testing
* Integration Testing:
* Equivalence Class Partitioning: if you test a small portion of a products, then it works fine, then it supposedly works fine the entire section. You take a sample of each which has similar characteristics and you assume which all of them works fine.
* Boundary Value Analysis:
  + Test always the minimum and maximum, you don’t have to pick the middle.
* Test Documentation:
  + Test Plan
  + Test Scenario
  + Test Case
  + Traceability
* When do you Automate?
  + You do automation once your environment is stable, you have to test it manually once, before you do automation test.
* When a new functionality comes, you have to manually test it, before you automate it and use from your skill/experience.

Example: open whatever browser you open, type www.Amazon.com , if it comes up then it is a manually testing.

* Every team is following assigns, one person will be production owner and then a couple of Web-Developers, we will have a business analyst.

**Software Development Lifecycle (SDLC):**

* Requirement gathering: Requirement is the initial step in SDLC and you gather it to develop an application.
* Design: Design takes places in the second stage by the software architecture engineers. The design phase comes after a good understanding of customers’ requirements, this phase defines the elements of a system, the components, the security level, modules, architecture and the different interfaces and type of data that goes through the system.
* Implementation or coding stage: Developers are developing the modules (DEV Environment), Unit testing happens by developers, they also deploy it to Quality Control (QC). (when the bottom of QC is active, then the testers start their project or work period, they usually say they are working in certain tickets like; login ticket, which is the term used by testers).
* Testing: It happens by the testers, (LOE level of efforts). Always put extra hours for testing to divide how to implement your testing.
* Testers will test with different scenario in test environment.
* Testers normally read the requirements first, then run the tests, if there were not requirements then we talk directly with the developers and come up with the different scenarios and we do test cases to pass it. We have to analyze the scenario before taking action, even if we don’t have requirement, you have to go to developers, business analysts and finally resolve the issue.
* The chain of command for taking action to resolve a bigger defect, usually the testers report to developers, if developers couldn’t solve it, then report goes to the business analysts and if business analyst think that it is important enough to be brought to the attention of product owners, then they have to ask permission or approval from the product owner for implementing certain functionalities (add or change something to the existing application).
* Example: verify that user should be able to login in signing in page on Amazon.

Updated AC:

Verify that user should be able to login sing in page Note: email box should contain valid email address. And password enter box should contain special character, alphabet and numbers.

* [www.portal-test.appian.com](http://www.portal-test.appian.com)
* Deployment:
* Maintenance:
* Functional Testing: (manual testing the best functionality of the application)

Teacher: Ms. Guljannet

**SDLC (Software Development Life Cycle):**

* Different between Validation and Verification: Verification happens during the process and Validation happens at the end of the project.
* Smart Criteria: The requirements

Software Development Life Cycle Overview:

* Different phases of SDLC: The software analysts will ask from the client requirement. Design phase is done by the software engineer architectures.

1. Requirement phase:

* What should be done in requirement phase:
  + Gathering the requirement:
  + Analyzing the requirement:
  + Documenting the requirement:

1. Once the requirement phase is done, then the requirements will be analyzed by the business analyst and will be sent to the design phase. The software architecture engineers will handle this phase and then they will send it to the software engineers for coding stage.

Example of the Restaurant:

* If you are a Restaurant owner and you want to have software assistance for creating an application which can implement certain functions:

1. The system or application should be able to track the employee’s hours of work.
   * + I want that my employee to use ID for Log-in:
     + Authorized user should be server, chef, manager, bartender, busboy
     + System should be tracking employee’s hours by daily.
     + System should be able to track the break time as well.
     + Employees should not be able to enter hours manually. (System admin can do that).
2. Servers should be able to order menu from the application
   * + System should be able to display the price of each item/food, when server select that certain item.
     + System should be able to calculate the discount.
     + Each order should have server’s ID number (meaning each server will have a numeric identification number which is identified by like a roster number which is in military Units).
     + Kitchen should have a printer which is connected & programmed directly from the ordering stage and the printer will automatically printout the order/ticket.
     + Server should be able to click on delivery or takeout option, dining.
     + The menu should include all types of foods and the menu should be categorized by appetizers, main entrees, drinks, side dishes, kids’ menu.
3. I need a report everyday so that I can able to evaluate my business how does it running and how many food orders sold per day and the total amount.
   * + I want to include date/time, total price of orders, total cost, taxes, tips, Grand-Total.
     + Detailed report is required to explain which food’s sale percentage high and which food percentage is low. Quantity of sale on daily basis.
     + I should be able to check the report anytime.

\*\*\* Functionality: is the purpose of application why we use an application.

\*\*\* The software architecture engineers who handles the design stage is like the architecture engineers who design a building, before you want to build a building you tell your idea to the architecture engineer and he will design it, gives you a prospective how would the building look like, then site engineer will implement that design to build a building. It can be a good example for software architecture engineers how they perform their responsibility.

\*\*\* Software developers will work in coding stage, when they start working, they work separately, and they push it to be applied in the application. They test it and they deploy it, but they will do an integration and regression test before deployment.

\*\*\* Testing Phase: We need to check each Unit, Component and Functionality to make sure it working as expected. We have to verify the requirement and make sure that it is working well. When the expected result is not matching the actual result, then it is a Bug.

* When you find a Bug, what is the process and how you handle it? You need to reproduce 2 to 3 times to make sure it is a Bug. There are 2 types of Bugs:

1. What is a valid Bug: you find an issue which is a functionality and codding issue that the developers have to fix it.
2. Invalid Bug; is a Bug which is a problem somewhere else and there is no issue with the coding or design issue. In this case the developers don’t need to take an action. Testers can fix the Bug.

Procedure of exploring a BUG:

* Reproduce 2 to 3 times to make sure it is an issue for sure.
* Take a screen shot or video if possible.
* Open the ticket:
* Steps for reproduces:
* Open URL
* Login as user with invalid email address expected result: system should display error message.
* Actual Result: system is not displaying error message.
* Test date: [gu\*\*\*@gmail.com](mailto:gu***@gmail.com)
* Test Environment should be identified for the software developers to fix it easily. The test environment should be copied and passed like; URL. [www.port-test1.come](http://www.port-test1.come)
* dev environment: [www.prot-dev.com](http://www.prot-dev.com), it is the development stage once you make sure there is an issue and it can be fixed by the developers.

\*\*\* The testers should have an excel sheet, ticket number, task-case number, assignments should be tracked, and it is very easy to track all your activities. You avoid losing the BUG tracking.

\*\*\* Once you open a BUG, you should assign it to the developers, developers will be looking in to and fix it as soon as possible. They reassign it to the testers and the testers will retest the BUG and make sure it is fixed. If it is fixed then the testers will close the BUG, if there as a problem again, then the testers will reassign the BUG to the developers to re-fix the issue (BUG).

Teacher: Mr. Nader

**Method which helps to build an Application:**

Waterfall: Still can happen inside of the process especially in some government jobs.

Waterfall is a process of one point to another, top to bottom, it is basically one step is completed and another step starts.

Waterfall starts with gathering requirement, before that the idea comes, in order to make this idea, you start gathering the requirements. In Waterfall once you finish a level, it is done, you can change the requirements after a while, but very costive, they rely on the requirement and design the application. Once the design is done, the development will start. The development process you just focus on development, no need to focus on requirement or design stages. Once the development is done then the testing starts. Once the testing is finished then the production will be ready to be released. Waterfall is still used mostly in military units.

In waterfall they spend as much time as possible to build an application and test it properly. Once it is released, then nothing will be changed.

Waterfall will be helpful for the military and government application.

What is the advantage of the waterfall: If you deal with very sensitive and high classified and security related application, then you can use from the waterfall. In waterfall you have enough time for everything and you do it very accurately.

The period of the work process in Agile is called Sprint: like; 1 week-sprint.

SDLC DEFFINATION: is start from the idea, gather the requirement is gathered, design is done, the codding takes place, then the testing stage starts, once the testing is done, the application will be released and, in the meantime, you have to focus on maintenance part.

Easy way to find job:

Our work will be Automation Testing Engineers, our job will be testing application to functions properly. In order to be able to successful in your interviews, now pick an application and suppose that you work for that Company. You start all the functionalities about that certain application, sign-in is functionality, sign-out is functionality and each functionality should be exercised most often to get familiar. Try to understand how these application work. Just prepare how to talk in the interview.

Group study is very helpful, we usually talk during the interview, knowing is not enough, you should be able to reflect your knowledge to the interviewers. You have to do knowledge sharing, study something for 30 minutes and explain it to your group members. Just focus on specific useful topics. Mentoring is very important too, after a while you have to switch gear and act like a mentor.

There are different ways to develop a software:

1. Agile
2. Waterfall
3. Scrum

Agile Manifesto:

These terms are very important to practice with, it is used during the job process and are professional.

Agile:

Agile is the method or way how to develop a software. Agile gives number one priority for customer, it is very important to be flexible for your customer and customer plays an important role. The customer can see the process of development. Agile is more like a team work, developers, testers, analysts, product owners all chat with each other to come up with a solution.

Agile:

1. is marked by ready ability to move with quick easy grace – an agile dancer.
2. Having a quick resourceful and adaptable character – an agile mind.

* Individuals and interactions: it is very important to do interactions, most defects are not real defect, the developers just mis-understood or sometimes was missing from him.
* Customer collaboration: In waterfall, the testers just give it to the developers and it is done. But in Agile, you interact everyday with the developers and customer is a part of the process from the beginning and the customer wants to see the development process. Sometimes the customer is not there, but they have representatives which are product owners and mostly they understand the software development process.
* Responding to change:

Principles of Agile:

1. Satisfy customer through early and continuous delivery: continuous delivery makes the customer very happy. When agile started it was 4 weeks sprint, they later came up with 2 weeks-sprint (sprint is just a time frame).
2. Agile welcomes changing requirements
3. Deliver working software frequently
4. Collaboration between the business people and developers
5. Build projects around motivated individuals
6. Face to face conversation
7. Primary measure of success is Working Software
8. Sustainable development = constant pace
9. Technical excellence and good design
10. Simplicity
11. Self-organizing teams
12. Reflection to tune and adjust team behavior

Non-Agile:

* Process and tools
* Comprehensive documentation
* Contact negotiation
* Following a plan:It is very important to follow the plan, it is usually match with agile prospective. You have a deadline and you must finish the plan before the deadline.

\*\*\* Before you go to interview, do some research to see who is interviewing you, that way you can better prepare yourself to get an offer.

User stories:

* As a user, I can back up my entire hard drive
* As a power user, I can specifically,

Use Story Life Cycle: (we will use it in the future).

* Request level

Epics:

* Epics are bigger user stories
* Add more details
  + By splitting a user story into multiple, smaller user stories
  + By adding “conditions of satisfaction”.

Release: whatever goes to the production is a release, sometimes it is production shakeout and sometimes it is the production only. In shakeout production it is a limited production and mainly used for initial testing purposes. It is after testing is done and goes to the production.

How often do you release? You usually finish your sprints, you save it, take the approval from the business analysts and then usually it gets released every 3 months or 4 times a year. You have to come up with a number or digit, just track the release updates like for 2015 the number 3, every 3 months up to the end of the year will be 3.1, 3.2, 3.3 and 3.4. next year 4.1 for the first quarter of 2016 and so on.

Even in each company different projects, the sprints are different, some do 2 weeks sprint and some other do 4 weeks sprint.

Teacher: Mr. Nader

**SCRUM:**

Automation Test Engineer: Job Description, tester with automation engineer and we use from Java to do it. Usually, loop and if statement and conditions are used in workspace. Try to do the exercising and complete as much as you can, if you couldn’t do it, then please google it and take the knowledge later on. During the interview, if you answer 75 % of the questions, then you are succeeded. But, try to answer 100 %.

Advantage of Waterfall:

* Waterfall is used for smaller projects
* Less complicated
* Clear requirement

Disadvantage of Waterfall:

* If the projects are complex and open to redesign, then don’t use from waterfall.
* Waterfall takes very long time to bring even a small change, because bringing changes to waterfall is costive and time consuming.

Agile Advantage:

* Open to change the application or open to add to the application. Agile, is easy to bring changes in the projects or Application.
* In Agile everything happens very fast, you can add and change pieces in the application very easily.

Example: if you remodel the kitchen, in Agile case, it is very easy to change every single part of the kitchen more often, without changing all parts of the kitchen, but in waterfall case, you have to change a lot in order to bring change a small part of the kitchen or an application.

**Scrum**:

* Scrum is a part of agile methodology; the Scrum Framework is product backlog --->
  + Product Backlog: The functionalities go to the product backlog, you can add as much functionalities as you want, as soon as you have enough money to accomplish it. Product backlog is a place where you store all the ideas. It is basically a storage for the requirements and if you change anything, you store it in your product backlog.
  + Sprint Backlog: is a timeframe that you work with your team to finish the project. Usually, the sprints are 2 weeks and it depends based on the projects. Usually, 2 weeks & 4 weeks are the usual timeframe used for sprint.

1. What happens if you promise to finish a project in a certain timeframe and you fail to do so? Answer: You must plan very accurately in order to avoid those types of situation, you have to be very precise as for as timeframe. It is better to avoid the failure of not completing the project in dedicated sprint timeframe.
2. Blocker; is a prevent to block your process of working and doesn’t allow you to deliver your project on time. You can rework on the application next week, you will not be fired for unintentionally blockages.
3. ScrumMaster:
4. Product Owner:
5. Team: (Developer, testers)
6. Product owner: the job for product owner is job is to sets with the person who has idea and translates it to the language of web developers. The product owner can have business analysts to work under him. Each team has business analyst and the product owner has bigger responsibility. Product owner makes sure that the entire teams do their work properly. The person with the idea can participate with product owner in all meetings. In small projects, the product owner can be a Business Analyst (B.A). Product owners presents the client. If you have a product owner in the project, he will communicate with the clients and product owner deals with business.
7. Development Team: It doesn’t mean how big the project is, it is always same structure. Developers deal with the application and to make sure that the application is working fine.
8. Scrum Master: Is organizing the project, meetings and identifying the problems to get resolve it from a business & development prospective. If a sprint is for 2 weeks, the Scrum Master will talk with the whole team to make sure that the project is completed and ready to be tested or released. The Scrum Master is basically making sure that the testers and developers do their job properly. Scrum Master coordinates on the behalf of the team with the other teams, sending emails, making phone calls in order to better accomplish the project.

Sprint Events or Scrum Ceremonies: If you say in interview that you worked in Agile methodology and you know all the Scrum Ceremonies, it will look good and it is basically the different meetings happens on daily basis.

* Sprint:
* Sprint Scrum:
* Daily Scrum:
* Sprint Review:
* Sprint Retrospective (it is like a recap):
  + - 1. Sprint Planning Meeting: happens at the beginning of a sprint, usually the sprint planning meetings happens on Tuesdays. The maximum of sprint planning is 2 hours, if it is 2-weeks sprint, the sprint planning meeting hours are 1 hour. If it is 4-weeks sprint, then it is a little bit longer which could go up to 2 hours. The sprint planning meeting you basically decide what you do as a tester or as a web-developers during the next upcoming 2 weeks.
* Who attends to the sprint planning meeting:
  + The Scrum Team takes part in the sprint planning meeting.
  + User story discussion / negotiation.
  + We talk about the acceptance criteria, make sure that the user stories are ready. It is the definition of done.
  + We do pointing: after each period of time we give points to the user stories like; 1, 2, 3, 5, 8. When user story is clear to the developers to the testers to test, then you give the application a pointing to describe how clear the application is. Pointing is to make clear the team capacity. If you give the points to the application, you have to make sure the total is not more than 20. (<=20). The product owner always point lower, the testers usually give higher points. When you give the points, you don’t give point as a tester only, you give the point as a whole team from user-story prospective.

Example:

1st user story - 3 points

2nd user story – 8 points

3rd user story - 8 points

4th user story – 8 points

(if it is like this, it is not good, as a tester you should pick another functionality with the lower points and start working on it).

Stretch Code is the term used for testers which is not their responsibility. If it passes or fails, it really doesn’t matter for the tester.

* + - 1. Upcoming Sprints Goals and priorities: product owner decides what is priority for the team.
      2. Discuss, negotiate, finalized:
      3. Definition of DONE / Definition of Ready: if we can work on user-story, it is definition of ready. We can write test scenarios based on that.

1. User’s Story should be ready
2. it has clear acceptance criteria
3. Data should be ready like username and password should be ready
4. For our team (testers), we should have testing environment or virtual machines should be ready. Each project can have different virtual machines.
   * + 1. Acceptance Criteria: If all the criteria’s match, then the user story is good. Valid username and password.

The user story criteria:

1. Valid username, valid Password ----> you can sign-in
2. Invalid username, invalid password ------> Error.
3. Etc ….
   * + 1. Standard of care:

Application: has 2 ends, front end and back ends. The front-end has the visuality and doesn’t do anything with the functionality. Some people draw in paper and the developers implement the functionality. The testers sometimes test the functionalities without visuality. We don’t test how it looks, we test how the application works.

Functionality: is set username, set password, once it goes through with the right username and password then it is a functionality. This is a part of the functionality and it could be hundreds of different functionalities. Functionality is the code which is written by the coders or developers.

User’s story: is a small functionality that can be developed in 1 a week. Every user’s story has an acceptance criteria. The business owner accepts the criteria.

The user story criteria:

1. Valid username, valid Password ----> you can sign-in
2. Invalid username, invalid password ------> Error.
3. Etc ….

Positive Tests: are business critical scenarios or positive scenarios.

Testers: you should ask as a tester the details of requirements for the user story criteria.

User Story should be ready, it should be clear for product owner and developer.

Teacher: Ms. Guljannet

**Software Test Life Cycle (STLC):**

* Interview Questions should be noted separately and practice individually or with the group which will be very helpful.

**Interview Question:**

* 1. Can you explain the regression testing process in your team? Or what is your regression cycle? Answer is explained below in regression test notes.
  2. Have you ever involved in test planning or created a test planning? The testers usually don’t create test planning, the project manager and the business Analyst create the the test planning.

Testing:

1. Dynamic Testing: it is testing application by executing codes. It is also called execution testing.
   1. Functional Testing:
      1. Unit Testing: is done by developers, it is testing of small codes by developers at the initial stages of developing a software application.
      2. Component Testing: is done by developers, code-based testing, while combining 2 Unit Testing with each other to make sure it is working fine and there is no issue. It is also called component testing.
      3. Integration Testing: Test after integrating multiple component if the system works fine or not. Sometimes, component and unit work fine, but when we run system it may not work.
   * Example: updating of phones, any company once develop an application, it is not perfect, it is being updated as time pass. Integration Testing is usually larger than the component and unit testing.
     1. User Acceptance Testing: is testing the application from a user’s prospective to make sure it is acceptable by clients and it is also called UAT (User’s Acceptance Test). Big companies have UAT team to run this type of test and they just test to happy pass the application.
        1. User Interface Testing: In this test they usually test the application as End User’s prospective (End User = Customer).
     2. Smoke Testing: is initial testing process exercised to check whether the software under test is ready/stable for further testing. Before putting the entire team to testing effort, the system needs to be stable enough or worth to test. (the automation should check everything before the users use it).
     3. System Testing: It happens before the release, it checks all features from A to Z to make sure everything is working fine.
     4. Ad-hoc Testing: Testing performed without planning and documentation- the tester tries to break the system by randomly trying the system’s functionality. It is performed by the testing team. It can be also called Exploratory testing. You explore the application to find bugs and other problems. It is also called monkey testing or Random testing.

* + 1. Regression Testing:
* Is from A-Z, every possible - case scenario about your application and make sure it is
* New feature, Modification, Update, Bug Fix, Anything Else
* Before we do release, we do regression test, before it goes to the public the regression is done. It is like full application test.
* The purpose of doing regression is to do as much as you can to prevent Bugs before the release.
* It is the last step of testers job to do regression test before the application is released for the public. Usually, every 3 months the release happen. The release happens during the night when the users are not using. If the application goes down for 20 minutes, the big application can cause loss of a lot of money. Example; Bank of America.
  + Regression, if any functionality is added we have to make sure the new functionality did not break existing functionality.
  + If Bug is fixed, then we have to make sure the bug did not break other functionality that was working fine. You have to run a regression test to make sure the whole system is working.
  + If there is any change in the code, we have to make sure that developer did not break the other codes unintentionally.
  + Scenarios are just each step.
  + Regression testing suite will have a lot of test cases. Some companies can have more than 2000 test cases in their projects.
  + Regression is repetitive and very time consuming. It could take 3 days, it could take a week.
  + Regression test cases are perfect for test automation. To save time and if we save time we save money.
* Major Regression: In a year you can have about 4 major regression mostly every 3 months and at the end of the year before Christmas you just run a final major regression to close up everything.
* Minor Regression: happens after each sprint to make sure all the functionalities that were developed during the sprint is working fine.
  + 1. Positive Testing:
    2. Negative Testing:
  1. Non-functional Testing: avoid saying this type of test in the interview, but it is checking the visibility and liability of software application.
     1. Performance Testing:
     2. Load Testing
     3. Usability Testing
     4. Reliability Testing
     5. Volume Testing
     6. Stress Testing
     7. Etc …

1. Static Testing: usually the developers do the static test which is reviewing the documentation and happens at the initial stage. Static test is done during verification stage.

* After creating test cases, as a tester, we review our each other’s test cases and it is peer-review.

1. Back end testing:

* Back end testing is referred to Database testing or server-side testing.
* The data entered in the front end will be stored in the back-end database.
* The database may be SQL server, MySQL, Oracle, DV2, etc. The data will be organized in the tables as record and it is used to support the content of the page.

Testing level /Hierarchy:

* Unit Tests: 1st happens. Usually done by developers.
* Integration Testes: second happens.
* End to End Tests: is considered system testing which is high level test. Third happens
* UI Tests: User’s Interface Testing and is the last stage of test which is done as a user’s prospective. Last happens:

**Software Testing Life Cycle (STLC):**

* STLC is a process followed by software industries to test the software products.
* STLC process is followed strictly to ensure quality products are delivered to the customers within the planned schedule and cost.
  + Requirement Analysis: The previous gathered requirements will be analyzed and also make sure what is testable or identify what will be tested. Scope definition, Create RTM, Automation Feasibility.
  + Testing Planning: Test Plan / Strategy, Resource, Estimation/Plan. As a tester we don’t have to create a test planning, the project manager is creating the test planning. The resource, budget, schedule is also planned by managers for the projects finalized the test plan. Automation tools are researched and selected (if automation is planned).
    - Test Plan: is usually done by the project manager, a document describing the scope, approach, resources and schedule intended test activities.
  + Test development Phase: Test Environment, Create Test Cases, Test Data, Test Scripts, Review test cases (peer review), create test data (if test environment is available).
  + Test Execution: Functional Testing, Smoke, System, Integration
  + Test Closure: Test Closure, Report, Test Metrics.
  + Retesting is to make sure that the previous found Bug is fixed and you retest.
  + Verify that user should be able to select any items from drop-down list. The instructor created a Task Case (created a ticket) and took screen shot of each item and pass that test. She was supposed to do manual testing that was very time consuming and she was able to pass the test case and later on automated the test case.
  + Jira: includes creating test cases, task, create bug. (these are functionalities that will be covered in Jira application). Some companies create task cases in excel sheet. In coming up classes it will be covered. Testers create the task cases and if you don’t know how to create task cases, then you will not be able to execute your job. Make sure that the task cases are clear and understandable.
  + We do 3 Amigo meetings in requirement analysis phase, 3 people join this meeting, 1. Business Analyst. 2. Tester & 3. Developers. The testers usually initiate or handle the Amigo meetings and the tester will write down an email to invite the Developers and Business Analyst.
  + Requirement Traceability Matrix (RTM): at the end they want to know which task cases covered and which task cases are not covered. Usually created in excel sheet.
  + Test Environment: it depends on how big the company or the project is, in current company of the instructor they have:
    - O & M team (Operation and maintenance team):
    - Enhancement team
    - UAT: UAT environment. They are working on functional testing.
* Test environment setup:
* Test Execution: before test execution environment should be ready.
  + - * Execute test cases
      * Document test results
      * Log defects for failed cases

**Bug Life Cycle:**

* The testers open the bug, the developers fix the bugs.
* What is the Bug Life Cycle? What are the steps for Bug Life Cycle?
  + - 1. If I find out a Bug, I have to make sure it is a Bug not the problem with the system. If confirmed.
      2. The next step is to open it in Jira.
      3. Explain the description, how it happened? How the Bug happened?
      4. Upload the Screen-Shot.
      5. The ticket is assigned to the developers. Once you assign to the developers 2 things happen.
         1. Maybe duplicated, something that is already found by another tester.
         2. Another reason is you can postpone it, if it is not a priority, then the developer will fix it later. (also called latten bug). You put it in Bug log and they fix it later on.
      6. When new Bug is opened, the status changed to Open.
      7. While the developers are fixing a Bug, he status of Bug will be changed as “development process”.
      8. After fixing the Bug, developers change the status of Bug as (QC).
      9. The testers will retest the Bug and if passed the test, then they will close the Bug.
      10. If not fixed, then the Bug will be reopened.

**Test-Cases:**

* For creating test case, we should have the following phases to be ready:

1. Test Case ID:
2. Pre-Condition:
3. Test Steps (Description):
4. Test Data:
5. Expected Result:
6. Actual Result:
7. Pass/fail:

* Some teams use from Excel Sheet to create test cases, some other IT teams use from Jirah to create test cases.
* While I am creating test cases, I have to open application in another screen to see the actual application and then create the test cases based on that.
* If you were not able to register, then there is a Bug and you need to open the Bug in Jirah.
* This was a happy pass or positive scenario
* while create test cases, if you find a bug, you have to open it, fix it and close it
* User Story: As a user I should be able to remove any item
* AC (Acceptance Criteria): Verify that user can remove any item from Cart
* While executing you can keep working on the test functionality to make sure that it passes the test scenarios
* If required saying enter valid user & password, then we should be clear about creating the test cases with details

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case Creating Method** | | | | | | |
| **Test Case ID** | **Pre-Condition** | **Test Steps (description** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| TC-01: | 1. Environment should be ready | Open the URL | https://www.etsy.com | System should display the home page successfully |  | Pass |
|  | 2. User should have valid Email Address to register | Click on "Register" link on top right of the page | email: khalili@gmail.com first name.Nematullah. Password: khalili786 | Once you click the register, System should display "Create Your Account Page" |  | Pass |
|  |  | Enter mandatory field |  | User should be able to enter all mandatory field successfully |  | Pass |
|  |  | Click on "Register" button |  |  |  | Failed |
|  |  | **Negative Testing:** |  |  |  |  |
|  |  | Open the URL |  |  |  |  |
|  |  | Click on "Register" link on top right of the page |  |  |  |  |
|  |  | Enter invalid email address |  | System should display a validation message saying "Please Enter a Valid Email Address": |  |  |
|  |  | Enter nothing in email address. First Name or password field |  | System should display validation message as "Can not be blank". |  |  |
|  |  |  |  |  |  |  |
| **Creating Test Case for Remove Functionality / Deleting Item from Cart** | | | | | | |
| **Test Case ID** | **Pre-Condition** | **Test Steps (description** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| TC-02: | 1. Environment should be ready |  |  |  |  | Pass |
|  | 2. User should have existing email address | Open the URL | [WWW.ETSY.COM](http://www.etsy.com/) | System should display home page |  | Pass |
|  |  | click on Sign in link on top right of the page |  | System should display sign in page |  | Pass |
|  |  | Click on Sing in link on top right of the page | [khalili@gmail.com](mailto:khalili@gmail.com) | System should navigate to user's landing page |  | Pass |
|  |  | Enter email address and password then click on sign in button |  | User should be able to select any item |  | Pass |
|  |  | After sign in, go to any department and select any item then add to cart |  | User should be able to add any item to the cart |  | Pass |
|  |  | Select the size and color and quantity from the drop down list and click on add to cart |  |  |  | Pass |
|  |  | Usershould be able to Click on "remove" |  | System should be able to remove any item from cart |  | Pass |

**Writing the bug report:**

**Project**: Reservation application

**Issue type:**  Bug

**Summary:** We were not able to reserve a room by using this Reservation Application

**Component:**  Reservation Application (hunt for spot page, Date)

**Description**:

Steps for reproduce:

1. We identified the Bug
2. We opened the Bug
3. We reproduced several times to make sure that it was a actual Bug
4. We took a screen shot from the Bug and let the entire QA team know that there was a Bug in our Reservation Application
5. We informed the developers about the Bug
6. We also attach a copy of the Bug to this report that way the developers and the rest of QA team members understand better about the Bug.

**Expected result:**

* The system should take me to registration page

**Actual result:**

* The system displays an Error message: "wow wow bully. You broke something, 500".

**Test Environment:**

* https://cybertek-reservation-qa.herokuapp.com

**Test Data:**

* Email: jalabaster7f@drupal.org, First Name: Teri, Last Name: Mapam
* 29/10/2018, 3:00 to 4:00

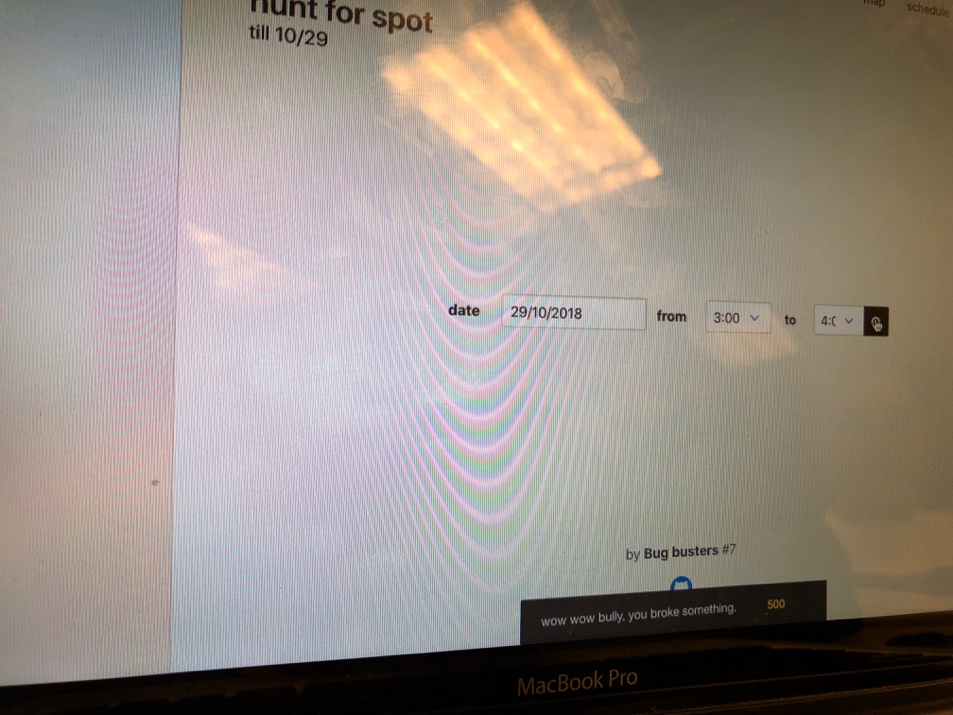
**Fix Version:** Current Sprint

**Priority:** Critical/Blocker/Major/Minor (Major Issue)

**Assigned:**  Development (Dev) Team

**Sprint:** Sprint 1

Bug Screen Shot:



Questions during interviews: Teacher: Mr. Moradil

* Phone screening
  + On the phone by either recruiting company or Client company
  + Normally it starts with “tell me about yourself”,
  + Talk about your day to day activities?
  + How to do you solve your challenges?
  + How do you learn automation?
  + then technical questions. It could be Java, selenium, cucumber or etc.

* Technical Questions in Java:
  + Technical questions can be different types, it could be Java programming, Automation related, different tools related. It could be Jenkins, git, etc. It include scenarios or frameworks. If there is java programming questions, they will start with asking about years of experience and how do you rate yourself from 1 to 10.
  + Other import questions is how do you use java for? The answered can be “I use Java for building automation framework, I am comfortable with core java, object oriented java, java collections, jdbc and etc. java collection is a part of core java, but better to be mentioned sperate.
* Then some specific questions comes including Algorithm of some questions? The questions will look like follow:
  + Array of ints, how to get unique values.
  + String and how to get unique chars and etc.
  + What is the difference between arrays and arrays list?
  + What is different between overloading and overwriting?
  + It could be anything around it.
* Phone screening usually ask couple of java, selenium, some cucumbers and some other technical tools that they are using.
* After passing phone screening, they schedule your interview face to face, it could be in-state or out of state.
* Sometimes it could be Skype interview and it will be the final interview. During face to face or Skype interview they ask technical questions to write codes and do certain automation tests. Sometimes, if the developers do the interview then the questions will be only focused on coding, because the developers don’t know about Automation.
* It is not only about technical site, they consider your personality, communication, knowledge of soft skills like; Agile, Waterfall, when did you start automating and sometimes, they don’t ask technical questions and it will be all soft skills.
* Once you are in final interview, then there will be a decision about you, if they accepts, then the HR or manager calls you and sends you an email saying that congratulation and then they start on your paperwork.
* They check also if you are a good fit, are you interested and then they will rate your skills. Always try to show interest in the project and they hire you.
* People can help u during your phone interview.
* There are some websites. Dice.com or monster.com. When u upload your resume there are HR guys will find u from these websites.
* Sometimes you couple of interviews in a week, sometimes once a week sometimes once 2 weeks and it really depends.
* If you know people in companies, then you can be hired quicker and you connection with the HR and employees in there.
* You can either look for the job vacancies through the websites or you will make yourself searchable that the companies can find you and reach you out.
* It usually takes time to create a good Resume.
* The following programming will be thought:
  + Java
  + Selenium
  + Web-driver/grid
  + Testing
  + Junit
  + maven
  + git+github
  + Cucumber bdd
  + Creating frameworks
  + AWS machines. Windows most probably, it is free for a few months and then will charge you based on your machine and it will be round 15 USD per month.
  + Automation:
    - Front end
    - Back end API
    - Database

* Once you have a job, then you will learn other methods that your framework is on, like c# and others.

=======================================================================================================

Date: November 1st, 2018

Teacher: Mr. Nadir Shafiyev

Subject: Jira

* How do you point in your project? In our team we have 1 point, 3 developers and 2 testers, we tell our points and the developers tell their points and then we justify the pointing to make sure how much time we need in total. As a tester you should say more points, then you can justify it with the team’s agreement.

* Usually, Stand-Up meetings are 10 to 15 meetings, if a problem is serious that can’t be handled in 15 minutes, then you just schedule that off site. It means that you go to your desk, open your calendar and you just arrange a meeting with that certain individual that you need to talk in detail.
* The work emails must be replied within 24 hours, if you are not available, please set a Auto Reply which let the email sender know that you are not available and you will reply the email once you get access to the computer.
* Test Scenario: When sprint starts you take part at the first day and you see the User’s Story, it is not developed, and you just see the user’s story. The developers will take a certain amount of time to develop that user story and during that time the testers will write their test scenarios. If it is log-in functionality, then you write scenarios like;
  + Log-in with valid credentials
  + Log-in with invalid credentials
  + Etc…

The testers write these scenarios to make sure that once you come up to the meeting you show your test scenarios and show them to the Business Analyst and other team members to see if they like it or maybe they add into your test scenarios. It is better you generate and write all possible test case scenarios that the BA and Dev & product owners like it and they don’t add anything on it. Then the Test Cases are created.

* Test Case:

Step 1: Expected Result Actual Result Date

Step2:

Step3:

* Test Script: is an actual code, the actual code you write, you keep all your Test Script in Eclipse.

**Project Management Tools:**

* If you have a project, then you need a management, there are a lot of tools to how to manage the project and managing the tests. Before, there were separate tools for managing the tests, now the Agile main job is to have one centralize tool to manage your projects and manage test in the meantime.

There are couple of software’s in the market that are in use for test management:

1. Jira: is working with open source tools like selenium, selenium is free and you can do basically everything that UFT does and it is free to use.
2. ALM: is an HP product, only works with license products.
3. UFT: is a testing automation tool. It is not free, you have to pay a lot of money per person to use it and it has license.
4. U-Track: is also free software, it is free and open source which does the same functionality as ALM and other test management tools do.
5. Leanft works with java, it is not free and requires payment to use. Leanft is very expensive to use.

OBJECTIVE:

Jira Basic:

* Why we use Jira? We create our sprints in Jira, once it is created it shows sprint backlog, testing, testing is done, all the speciation of the sprint and bug life cycle. We see each story’s phases through Jira every morning before we start our job. We usually updated our tasks on Jira Board to brief on Scrum Meeting on daily basis. As a tester we have tasks, we need to update the Jira your tasks to explain at the Scrum Meeting.
* Search and Reporting:
  + Scrum Master likes to have the Search and Reporting part, he wants to know each team member what they accomplished during the last year and needs an detailed report.
  + Usually the testers or developers write a task case and give a point for even very basic and easy jobs in order to make the final yearly report to the Scrum Master that he accomplished a lot. It is important that you show how effective role you played.
  + At work each team member has their username and their own level of access and certain works are done by taking official written or verbal approval from the management.
  + Search by functionality: you can filter your search by different functionality, we usually focus on the testing scenarios.
  + Search by user’s story
  + Powerful and flexible searching
  + Support searching customized field too.
* Customizable Work Flow and Fields:

Issues Type:

* + Testing Bug:

Work Flow: It is basically a Bug Issue Life Cycle and takes the following steps:

* + Open:
  + in progress:
  + Resolved:
  + Closed:
  + Reopen:

Field and Screen: This is a format of assigning task to a team member as below:

* + Issue Type: Sub-Task!!!
  + Priority: Major, usually the major releases are going by numbers like major1 , 2, 3 …, the minor issues are go like 1.1, 1.2, 1.3 and etc.
  + Severity: None
  + Bug Type: None
  + Test Session Type: None
  + Component/s:
  + Fix Version/s:
  + Attachment:
  + Assignee:

Screen of assignee: It will include the following information:

(Unit Test Module A):

* + Type:
  + Priority: Actual Version:
  + Components: Project1
  + Labels: None
  + Description: Click add description.

Resolve Task:

\*\*\* Usually we do not add any point at the middle of the sprint, it effects your pointing and ideally all points are done in sprint meetings.

\*\*\* Differ Issue: you can assign an issue to a developer who has not developed the code and you prefer to assign the Bug or issue to another developer.

* How to work in Jira:
  + Task Management:
  + Bug Management:
* Usually, the tests are not accomplished by the person who writes the code, the peer review or assigning a colleague to do that. One user’s story goes to 20 different testers to make sure that the certain user story is functioning properly and pass all test cases.
* There are 2 scenarios once you find a Bug, it is sent to the developers, either they accept or don’t accept. If they don’t accept, then they will send the Bug back to the tester. If developers accept a Bug, they fix the code and send it back to you (testers). The testers retest it and make sure it is properly fixed to pass the test. If it is not fixed, the tester reopens the Bug and send it to the Developer. If you don’t have enough time a tester to test all test scenarios, then you have to document and bring it up to the BA attention that you couldn’t do all test cases based on the lack of time.
* There are Help Centers or Help Desk in each company and they can fix your technical issues by contacting them via email or physically. Mostly, government companies are run sensitive projects and you will not be able to copy or download their secure information.

Dash Board (in Jira):

* A dashboard is a collection of gadgets that display information.
* Burn-Down Chart: It explains that will you accomplish during the upcoming sprint. If you are able to finish your part in coming up sprint period.

Complete Tasks, Remaining Effort, Ideal Burndown, Remaining Tasks.

Jira Concepts:

* + Collection of Issues, then the issues go through the work flow (Open, in progress, Resolved, Closed, Reopen).

* + Create Task, Do Task, Resolve Task, Close Task then the same cycle (Open, in progress, Resolved, Closed, Reopen).
  + In big companies, there is a Jira administration team and it is very important to handle the Jira side.

Task Management Process:

* + Scrum Master: Assign usually task to the team members, but each team member can assign a task to himself.
  + Project team:
* Scrum Master Exam:
  + 2 different ones, one is expensive, easy to pass, another is cheap, but easy to pass.
* OCA Exam:
  + Take the java knowledge and in the future, we will get the certificate by taking 65% score.
* AWS:
  + Is important and easy to take it. You have to know the test cases of AWS, then you can take the certificate later on.

Helpful Youtube Video about Jira:

* X-ray for Jira:
* Xray - Test Management inside JIRA (Extended Demo)
* https://www.youtube.com/watch?v=VbisAffbKfs

==============================================================================

Date: November 8, 2018

Teacher: Mr. Nadir Shafiyev

Question:

* What is the list of SDET & documentation to have at real work environment?

It is better not to have more documentation, all documentation should be in Jira, there are user stories, test class, test cases and everything will be stored in Jira. We don’t need extra documentation, in Agile the adjustment is done according the team and usually we need the following documentation:

1. Agile Release Documentation: When our story and Agile story is done, then we complete this documentation. You can fill these documentations in your major release or minor release, it belongs to you.
2. Developers => Agile Development Documentation: It belongs to the Developers, but it is inside the Agile Release Documentation. The developers fill how many user stories they developed, and they document it and give information about it.
3. Testers => fill the Agile Test Documentation or Document which put all the information about the Agile Release Documents.

* Usually, there are templates at work that you fill, you don’t write everything from the scratch and it is easy to learn if you read carefully.
* The type of your sprint will explain how mature Agile you are, if you have 1 week Agile, then your level of maturity is Best. If you have 4 weeks sprint, then you are not that much good in Agile. If your team does fully Automation Testing, then your team is mature. Manual testing is also important, but the more tests you Automate the more mature your team is.
* Automated Deployment: While developers finish the code, they push it, then the testers will test it. When you learn Jinkins, then the Automated Deployment will apply, and you can get better knowledge if it. Codes deploy from development environment to testing environment.
* Jira & Jira x-ray: it is the same, x-ray is just an extension to Jira.
* Common Pointing: It prevents from the additional work, you point the testers & developers in the same application in Jira which make it easier and prevent from writing additional documents.
* As a tester, first we do manual testing, if you can’t do manual testing, then you can’t Automation. The application that you want to work in the future, try to find the defect and send it to them, maybe those companies give you an offer to work with them in the future. We have to be very good manual testers and catch problems.

Testers’ main jobs:

* Web Apps => User’s Interface Application. We are testing Web Apps. (Slack is Web Apps, Desk Apps and Mobile Apps).
* We can test back end. Back-end is a database. Whenever you post a picture in Facebook, then it goes to the Facebook database. We can test it.
* The testers also test API like Zabiha Application which gets direction from Google Map.
* The testers create test plans for User’s Story. Our test plans should complete the acceptance criteria.
* Usually, the developers and testers have issue about the pointing. If the tester gives more points to the developers or the testing team, then the product owner will be frustrated, because the product owner is pushing the testers and developers to get less points to do more user stories.

Jira Software:

* What we care about projects? In big enterprises, everything is designed abased on your user stories.
* In sprint planning meeting, the tester’s job is to choose a user story and move it to sprint backlog. First day of the sprint the sprint meeting happens, testers, scrum masters, developers, product owner get together, and the product owner shows some of the priorities and then they start move it to sprint backlog. Once you get the user story and move it from product backlog to sprint backlog then you point it and it is the benefit of Agile that you receive a User Story first, then you point it based on the capacity of the work.
* Scrum Master is responsible for creating sprint, closing sprint and he is the one who moves the user story from the product backlog to sprint backlog.

Jira Functionalities:

* Sprint Backlog: it means our sprint planning
* Construction Analysis
* Construction Done
* Testing
* Done
* As a tester when I go to my job in the morning, I look at the Jira and see what is done, what needs to work on and if something is almost done, then I will try to complete it and move it the done Column list.
* If a user story is in product backlog, it means it is a kind of ready to be developed.
* What is closing a user story? It means to finish your job or responsibility that was given to you during the Sprint Meeting. It is also possible that you already finish your user story and you need it to Testing Colum that the testers test for the final pass. In some companies only, product owners and Business Analysts accept or satisfy, and they approve it to be moved to the next step or done Column. Sometimes, once you do your user’s story, then it will be automatically removed from your product backlog to the next column, but sometimes you have to remove it yourself after the approval of the product owner or scrum muster.
* Every team has different norms, sometimes the Scrum Master has bigger role who is approving the moving user story from one Column to another (from product backlog to testing or done backlog). Some companies, you need to demo, once the whole leadership accept or approve it, then they move it to the done Column. The demo is usually done in test environment and it is done by the testers. The testing environment must be product environment. Example Facebook. When you go to Facebook testing environment you will see everything that you see in a Facebook page, but all of them are mocked (fake).
* The first day of the sprint will look like everything is in Product Backlog, then day by day the user story will be moved to different columns like Construction Analysis, Construction Done, Testing or Done. At the end of the sprint, when the demo and retro is done, then all user stories will be moved to Done Column.
* In demo, you open the URL. As a user story when I go to URL, I type the BOFA.com, the Bank Of America should open. In demo, you open the created or made functionality or application and you just show it (you demo).
* Question: whenever there is a release, you just create another specific sprint for 1 week or 2 weeks and it depends how big your release is. In that sprint you just create tasks. Usually, when it is release time, the whole team don’t work on other product backlogs and the focus to support the release.
* Jira: it is usually updated every day at the daily stand up meeting. When you find a defect, then you assign it to developer directly, no need to ask the scrum master again, you just send an email to the developer and you assign the user story to the developer.
* The best chart is Burn-Down chart which is a graphical representation of work left to do versus time. It is just for seeing your progress, it is easy to see the overall picture of your productivity. You can see how much you did and how much you can support your job in the future.

Creating a user story:

* Epic is created in product backlog. Epic is bigger user story, usually in mature Agile we don’t take Epic, we take big user story to make Epic. Good product owner assign Epic to the user story.
* Difference between a task and user story: task for individuals, user story is for the team and you do a demo.
* Acceptance Criteria is put in the information table while creating a User Story in Jira. The following Acceptance Criteria can be put in User Story:
  + When the user sends valid credentials, personal page should appear.
  + When wrong credential entered, valid error message should appear.
* Sprint planning meetings are handled by the scrum master, then the scrum master is updating the acceptance criteria, but the scrum master talks with the testers and developers about the accepting criteria while updating. The main issue starts once the pointing started, the testers and developers always want more point to take time for better performance, but the product owner and scrum master always want more points.

===========================================================================

Date: November 15, 2018

* Is Jira compatible with Agile? Jira has a good reporting system and it is compatible with Agile and it demonstrate your progress during the sprint. It also shows all Scrum ceremonies and you can see the sprint backlogs and all those different steps of Scrum in Jira easily.
* We can manage our project, but we cannot manage our test process in Jira. If we have Jira X-ray which is an add in Jira, it allows you to manage your test process in Jira.
* Jira Administration is a team, you don’t need to be physically in same office, but you are connected and the whole team can follow each other’s work progress. Sometimes, it is restricted, you can’t see each other’s work progress. Jira Administration team also responsible for any technical issue and that is why it is a paid tool.
* If you don’t know how to use Jira, it could crash, because you should not create multiple user stories for no reason and you have to know how to use Jira in a right way.
* You can use Jira in Waterfall and it is a tool that can be used both Agile & Waterfall.
* Every Single team has its own way of implementation and the sprints are different, maybe, 2 days, 2 weeks, 1-week sprints. But the general ideas are followed by all teams.
* For maintenance reason, you may receive email that during the weekend Jira is not working, but it not a blocker.
* Lower Environment: Where we develop and test, it is a lower environment. It means the application has production in lower environment. Lower environment is that with mocked data, you need test the application like; Facebook. You have to create a fake or mocked user and password to get access and further test out the application. If you don’t have access to lower environment, then it is a blocker.
* Different between verification & validation? You verify the steps before doing the action and validation is the result of your action which comes later.

Date: November 1st, 2018

Teacher: Mr. Nadir Shafiyev

Subject: Jira

* How do you point in your project? In our team we have 1 point, 3 developers and 2 testers, we tell our points and the developers tell their points and then we justify the pointing to make sure how much time we need in total. As a tester you should say more points, then you can justify it with the team’s agreement.

* Usually, Stand-Up meetings are 10 to 15 minutes, if a problem is serious that can’t be handled in 15 minutes, then you just schedule that off site. It means that you go to your desk, open your calendar and you just arrange a meeting with that certain individual that you need to talk in detail.
* The work emails must be replied within 24 hours, if you are not available, please set an Auto Reply which let the email sender know that you are not available and you will reply the email once you get access to the computer.
* Test Scenario: When sprint starts you take part at the first day and you see the User’s Story, it is not developed, and you just see the user’s story. The developers will take a certain amount of time to develop that user story and during that time the testers will write their test scenarios. If it is log-in functionality, then you write scenarios like;
  + Log-in with valid credentials
  + Log-in with invalid credentials
  + Etc…

The testers write these scenarios to make sure that once you come up to the meeting you show your test scenarios and show them to the Business Analyst and other team members to see if they like it or maybe they add into your test scenarios. It is better you generate and write all possible test case scenarios that the BA and Dev & product owners like it and they don’t add anything on it. Then the Test Cases are created.

* Test Case:

Step 1: Expected Result Actual Result Date

Step2:

Step3:

* Test Script: is an actual code, the actual code you write, you keep all your Test Script in Eclipse.

**Project Management Tools:**

* If you have a project, then you need a management, there are a lot of tools to how to manage the project and managing the tests. Before, there were separate tools for managing the tests, now the Agile main job is to have one centralize tool to manage your projects and manage test in the meantime.

There are couple of software’s in the market that are in use for test management:

1. Jira: is working with open source tools like selenium, selenium is free and you can do basically everything that UFT does and it is free to use.
2. ALM: is an HP product, only works with license products.
3. UFT: is a testing automation tool. It is not free, you have to pay a lot of money per person to use it and it has license.
4. U-Track: is also free software, it is free and open source which does the same functionality as ALM and other test management tools do.
5. Leanft works with java, it is not free and requires payment to use. Leanft is very expensive to use.

OBJECTIVE:

Jira Basic:

* Why we use Jira? We create our sprints in Jira, once it is created it shows sprint backlog, testing, testing is done, all the speciation of the sprint and bug life cycle. We see each story’s phases through Jira every morning before we start our job. We usually updated our tasks on Jira Board to brief on Scrum Meeting on daily basis. As a tester we have tasks, we need to update the Jira your tasks to explain at the Scrum Meeting.
* Search and Reporting:
  + Scrum Master likes to have the Search and Reporting part, he wants to know each team member what they accomplished during the last year and needs an detailed report.
  + Usually the testers or developers write a task case and give a point for even very basic and easy jobs in order to make the final yearly report to the Scrum Master that he accomplished a lot. It is important that you show how effective role you played.
  + At work each team member has their username and their own level of access and certain works are done by taking official written or verbal approval from the management.
  + Search by functionality: you can filter your search by different functionality, we usually focus on the testing scenarios.
  + Search by user’s story
  + Powerful and flexible searching
  + Support searching customized field too.
* Customizable Work Flow and Fields:

Issues Type:

* + Testing Bug:

Work Flow: It is basically a Bug Issue Life Cycle and takes the following steps:

* + Open:
  + in progress:
  + Resolved:
  + Closed:
  + Reopen:

Field and Screen: This is a format of assigning task to a team member as below:

* + Issue Type: Sub-Task!!!
  + Priority: Major, usually the major releases are going by numbers like major1 , 2, 3 …, the minor issues are go like 1.1, 1.2, 1.3 and etc.
  + Severity: None
  + Bug Type: None
  + Test Session Type: None
  + Component/s:
  + Fix Version/s:
  + Attachment:
  + Assignee:

Screen of assignee: It will include the following information:

(Unit Test Module A):

* + Type:
  + Priority: Actual Version:
  + Components: Project1
  + Labels: None
  + Description: Click add description.

Resolve Task:

\*\*\* Usually we do not add any point at the middle of the sprint, it effects your pointing and ideally all points are done in sprint meetings.

\*\*\* Differ Issue: you can assign an issue to a developer who has not developed the code and you prefer to assign the Bug or issue to another developer.

* How to work in Jira:
  + Task Management:
  + Bug Management:
* Usually, the tests are not accomplished by the person who writes the code, the peer review or assigning a colleague to do that. One user’s story goes to 20 different testers to make sure that the certain user story is functioning properly and pass all test cases.
* There are 2 scenarios once you find a Bug, it is sent to the developers, either they accept or don’t accept. If they don’t accept, then they will send the Bug back to the tester. If developers accept a Bug, they fix the code and send it back to you (testers). The testers retest it and make sure it is properly fixed to pass the test. If it is not fixed, the tester reopens the Bug and send it to the Developer. If you don’t have enough time a tester to test all test scenarios, then you have to document and bring it up to the BA attention that you couldn’t do all test cases based on the lack of time.
* There are Help Centers or Help Desk in each company and they can fix your technical issues by contacting them via email or physically. Mostly, government companies are run sensitive projects and you will not be able to copy or download their secure information.

Dash Board (in Jira):

* A dashboard is a collection of gadgets that display information.
* Burn-Down Chart: It explains that will you accomplish during the upcoming sprint. If you are able to finish your part in coming up sprint period.

Complete Tasks, Remaining Effort, Ideal Burndown, Remaining Tasks.

Jira Concepts:

* + Collection of Issues, then the issues go through the work flow (Open, in progress, Resolved, Closed, Reopen).

* + Create Task, Do Task, Resolve Task, Close Task then the same cycle (Open, in progress, Resolved, Closed, Reopen).
  + In big companies, there is a Jira administration team and it is very important to handle the Jira side.

Task Management Process:

* + Scrum Master: Assign usually task to the team members, but each team member can assign a task to himself.
  + Project team:
* Scrum Master Exam:
  + 2 different ones, one is expensive, easy to pass, another is cheap, but easy to pass.
* OCA Exam:
  + Take the java knowledge and in the future, we will get the certificate by taking 65% score.
* AWS:
  + Is important and easy to take it. You have to know the test cases of AWS, then you can take the certificate later on.

Helpful YouTube Video about Jira:

* X-ray for Jira:
* Xray - Test Management inside JIRA (Extended Demo)
* https://www.youtube.com/watch?v=VbisAffbKfs

=====================================================================

Date: November 8, 2018

Teacher: Mr. Nadir Shafiyev

Question:

* What is the list of SDET & documentation to have at real work environment?

It is better not to have more documentation, all documentation should be in Jira, there are user stories, test class, test cases and everything will be stored in Jira. We don’t need extra documentation, in Agile the adjustment is done according the team and usually we need the following documentation:

1. Agile Release Documentation: When our story and Agile story is done, then we complete this documentation. You can fill these documentations in your major release or minor release, it belongs to you.
2. Developers => Agile Development Documentation: It belongs to the Developers, but it is inside the Agile Release Documentation. The developers fill how many user stories they developed, and they document it and give information about it.
3. Testers => fill the Agile Test Documentation or Document which put all the information about the Agile Release Documents.

* Usually, there are templates at work that you fill, you don’t write everything from the scratch and it is easy to learn if you read carefully.
* The type of your sprint will explain how mature Agile you are, if you have 1 week Agile, then your level of maturity is Best. If you have 4 weeks sprint, then you are not that much good in Agile. If your team does fully Automation Testing, then your team is mature. Manual testing is also important, but the more tests you Automate the more mature your team is.
* Automated Deployment: While developers finish the code, they push it, then the testers will test it. When you learn Jinkins, then the Automated Deployment will apply, and you can get better knowledge of it. Codes deploy from development environment to testing environment.
* Jira & Jira x-ray: it is the same, x-ray is just an extension to Jira.
* Common Pointing: It prevents from the additional work, you point the testers & developers in the same application in Jira which make it easier and prevent from writing additional documents.
* As a tester, first we do manual testing, if you can’t do manual testing, then you can’t Automation. The application that you want to work in the future, try to find the defect and send it to them, maybe those companies give you an offer to work with them in the future. We have to be very good manual testers and catch problems.

Testers’ main jobs:

* Web Apps => User’s Interface Application. We are testing Web Apps. (Slack is Web Apps, Desk Apps and Mobile Apps).
* We can test back end. Back-end is a database. Whenever you post a picture in Facebook, then it goes to the Facebook database. We can test it.
* The testers also test API like Zabiha Application which gets direction from Google Map.
* The testers create test plans for User’s Story. Our test plans should complete the acceptance criteria.
* Usually, the developers and testers have issue about the pointing. If the tester gives more points to the developers or the testing team, then the product owner will be frustrated, because the product owner is pushing the testers and developers to get less points to do more user stories.

Jira Software:

* What we care about projects? In big enterprises, everything is designed abased on your user stories.
* In sprint planning meeting, the tester’s job is to choose a user story and move it to sprint backlog. First day of the sprint the sprint meeting happens, testers, scrum masters, developers, product owner get together, and the product owner shows some of the priorities and then they start move it to sprint backlog. Once you get the user story and move it from product backlog to sprint backlog then you point it and it is the benefit of Agile that you receive a User Story first, then you point it based on the capacity of the work.
* Scrum Master is responsible for creating sprint, closing sprint and he is the one who moves the user story from the product backlog to sprint backlog.

Jira Functionalities:

* Sprint Backlog: it means our sprint planning
* Construction Analysis
* Construction Done
* Testing
* Done
* As a tester when I go to my job in the morning, I look at the Jira and see what is done, what needs to work on and if something is almost done, then I will try to complete it and move it the done Column list.
* If a user story is in product backlog, it means it is a kind of ready to be developed.
* What is closing a user story? It means to finish your job or responsibility that was given to you during the Sprint Meeting. It is also possible that you already finish your user story and you need it to Testing Colum that the testers test for the final pass. In some companies only, product owners and Business Analysts accept or satisfy, and they approve it to be moved to the next step or done Column. Sometimes, once you do your user’s story, then it will be automatically removed from your product backlog to the next column, but sometimes you have to remove it yourself after the approval of the product owner or scrum muster.
* Every team has different norms, sometimes the Scrum Master has bigger role who is approving the moving user story from one Column to another (from product backlog to testing or done backlog). Some companies, you need to demo, once the whole leadership accept or approve it, then they move it to the done Column. The demo is usually done in test environment and it is done by the testers. The testing environment must be product environment. Example Facebook. When you go to Facebook testing environment you will see everything that you see in a Facebook page, but all of them are mocked (fake).
* The first day of the sprint will look like everything is in Product Backlog, then day by day the user story will be moved to different columns like Construction Analysis, Construction Done, Testing or Done. At the end of the sprint, when the demo and retro is done, then all user stories will be moved to Done Column.
* In demo, you open the URL. As a user story when I go to URL, I type the BOFA.com, the Bank Of America should open. In demo, you open the created or made functionality or application and you just show it (you demo).
* Question: whenever there is a release, you just create another specific sprint for 1 week or 2 weeks and it depends how big your release is. In that sprint you just create tasks. Usually, when it is release time, the whole team don’t work on other product backlogs and the focus to support the release.
* Jira: it is usually updated every day at the daily stand up meeting. When you find a defect, then you assign it to developer directly, no need to ask the scrum master again, you just send an email to the developer and you assign the user story to the developer.
* The best chart is Burn-Down chart which is a graphical representation of work left to do versus time. It is just for seeing your progress, it is easy to see the overall picture of your productivity. You can see how much you did and how much you can support your job in the future.

Creating a user story:

* Epic is created in product backlog. Epic is bigger user story, usually in mature Agile we don’t take Epic, we take big user story to make Epic. Good product owner assign Epic to the user story.
* Difference between a task and user story: task for individuals, user story is for the team and you do a demo.
* Acceptance Criteria is put in the information table while creating a User Story in Jira. The following Acceptance Criteria can be put in User Story:
  + When the user sends valid credentials, personal page should appear.
  + When wrong credential entered, valid error message should appear.
* Sprint planning meetings are handled by the scrum master, then the scrum master is updating the acceptance criteria, but the scrum master talks with the testers and developers about the accepting criteria while updating. The main issue starts once the pointing started, the testers and developers always want more point to take time for better performance, but the product owner and scrum master always want more points.

===========================================================================

Date: November 15, 2018

Teacher: Mr. Nadir Shafiyev

* Is Jira compatible with Agile? Jira has a good reporting system and it is compatible with Agile and it demonstrate your progress during the sprint. It also shows all Scrum ceremonies and you can see the sprint backlogs and all those different steps of Scrum in Jira easily.
* We can manage our project, but we cannot manage our test process in Jira. If we have Jira X-ray which is an add in Jira, it allows you to manage your test process in Jira.
* Jira Administration is a team, you don’t need to be physically in same office, but you are connected and the whole team can follow each other’s work progress. Sometimes, it is restricted, you can’t see each other’s work progress. Jira Administration team also responsible for any technical issue and that is why it is a paid tool.
* If you don’t know how to use Jira, it could crash, because you should not create multiple user stories for no reason and you have to know how to use Jira in a right way.
* You can use Jira in Waterfall and it is a tool that can be used both Agile & Waterfall.
* Every Single team has its own way of implementation and the sprints are different, maybe, 2 days, 2 weeks, 1-week sprints. But the general ideas are followed by all teams.
* For maintenance reason, you may receive email that during the weekend Jira is not working, but it not a blocker.
* Lower Environment: Where we develop and test, it is a lower environment. It means the application has production in lower environment. Lower environment is that with mocked data, you need test the application like; Facebook. You have to create a fake or mocked user and password to get access and further test out the application. If you don’t have access to lower environment, then it is a blocker.
* Different between verification & validation? You verify the steps before doing the action and validation is the result of your action which comes later.

Example:

Create Issue:

* Project: Sample Project (SP)
* Priority: Hight
* Summary: Sample Xray Testing Sample Xray Test Plan (Usually you put the name of testing).
* Linked Issue: tests
* Creating test plan: Test plan is like a folder, test plan in x-ray hold the files about a specific user story and everything relevant to that.
* Description: This Test Plan is designed to test the Sample Xray User Story.
* Attachment: For test plan usually no attachment. We can attach the test plans or result either here or at the test plan.

**User story: Test Scenarios (TS)**

1 Sample xray xray1

2 Sample xray xray 2

3 Sample xray xray 3

* Pre-Condition: This test is not associated with any pre-condition

**Steps Data Expected Result Attachments**

1 Open url [www.facebook.come](http://www.facebook.come) The page should be available

2 enter credential Username & Password Homepage should be seen

* You can upload the result of your test in evidence section, if your test is passed, take a screen shot and upload it in evidence section. If you don’t see any result, try to reupload the page and then get whatever message is shown as a screen shot and upload it in evidence section in order to show for the rest of the team what happened during the test execution.
* The idea of Jira is that whatever is going on in the project, it is visible on the board and everyone in the team knows where we are standing as for as our work status.
* You can have different types of user story on each functionalities, like; user functionality, log-in functionality and etc.
* Epics are big user story, inside each Epic we have like 20 user stories.
* When you create the test plan, you don’t have anything in your Jira other issues column, once you started working on it, you will move it to the different stages like; Construction Analysis, Construction Done, Testing and finally Done.
* Acceptance Criteria is located inside the User Story. Each user story has it’s own acceptance criteria and one user story acceptance criteria will not match to another user story acceptance criteria. When you want to see your test plans even after 5 sprints, if you click on the linked issue and it will take you directly to that certain test plan.
* Definition of Done is like a checklist, 1. It should be tested 2. It should be developed another environment and a different person. 3. Business should accept it.
* Definition of ready: As a team, we need to make sure that the acceptance criteria is there, you give pointing and then it will be used for implementation.
* Test Sets: is a kind of helping us to put together scenarios together to execute. It is making our work easy and more organized give you access to see all test cases and details of it.
* Testing means execution, once the creating test is done, then you create execution and all of them you use xray in Jira to create.
* Jira cannot do automated test, you create it in Jira and the automation is done in Selenium. You can use other application whatever your framework is supporting, and you can run automation tests in your framework. Now, there are some ideas to provide a tool that be able to accomplish both manual and automation together.

Date: November 29th, 2018

Sprint planning meeting in real job environment:

User Story 1: As a user I should be able to make a reservation on reservation application.

Acceptance Criteria:

* Verify that ONLY the team lead should be able to make a reservation
* Verify that User can reserve only one room maximum per day.
* Verify that User can cancel the reserved room at any time.
* Verify that during the reservation, User can only book the room for passed time/date.
* Verify that user can reserve only for 2 hours max
* Verify that user sees available spots
* Verify the confirmation page
* Verify that the user gets confirmation email

* For pointing reasons, Scrum Master provides cards that are written 1 – 8 and you just use those cards for pointing reasons.
* Before asking questions, you need to read the requirements very carefully, then respond the question.
* Product owner is responsible for the user-story in general, acceptance criteria is a part of the user-story and directly handled by the product owner.
* In Agile, 1 project can have multiple teams and it depends on how big the project is.
* Zero Sprint, it makes scenes that when you start everything from the scratch.
* Test Framework: The combination of tools that you automate the test case, like; Eclipse, Java, Maven and extra testing like Junit, Cucumber and etc. The famous frameworks in market is Selenium, Cucumber and Maven. Intellij and Java are part of the framework.
* If your application getting data from another application, it is called upstream application and doesn’t matter if it is internal or external.
* Agile mature in testing field is if you use 100 % automation.

=====================================================================

Resume:

* Going on Market: to officially put your resume in market or get introduced to the recruiters.
* Cybertek Resume: Cybertek experience and education
* Traditional Resume: Traditional is that you put extra experience
* Things you can change:
  + Company name and experience can’t be changed
* Name, contact information and residency status is important for your resume

=====================================================================

**Date: December 13, 2018**

1. Question from Doc File
2. Jira
3. API / Manual API Testing
4. Postman

Do you use Jira in your project? Yes, we use Jira in our project.

Jira with Adds on:

Jira without Adds on:

X-ray:

* Test Plan for each user story
* Why do you need test plan?
* Test Sets: is a kind of helping us to put together scenarios together to execute. It is making our work easy and more organized give you access to see all test cases and details of it.
* Tests in Jira:
  + Manual
  + Automation:
    - Generic
    - Cucumber
* Test Execution:
  + Usually, I make test execution inside the test plan
* User Story: According to the User Story I create Test Plan, then Tests and Test Execution.
* Jira X-ray gives you an ability to write test scenarios and drug it to Intellije or Eclips and prevent rewriting the scenarios.
* One Test Execution can handle unlimited tests plans.

**What is API (Application Programming Interface)?**

* Communication between different applications
  + Postman is one of the tools, you use
* API is the acronym for Application Programming Interface, which is a software intermediary that allows two applications to talk to each other. Each time you use an app like Facebook, send an instant message, or check the weather on your phone, you’re using an API.
* API is very good tool to use for testing purposes, you can use in an Automated way to Automate certain test cases.
* What layers your application has?
  + It has User’s Interface (UI) / GUI Graphic User Interface
  + It has API (Application Programming Interface)
  + It has Database
* API: The term API is an acronym, and it stands for “Application Programming Interface”.
* APIs Make Life Easier for Developers
* APIs are used for communication between services
* APIs controls access to resources.

Imagine you are developers:

* Each application has its own API and you can talk to it through HTTP Request
* UI (User Interface) testing is that you actually put the link
* Download www.getpostman.com for API test purposes

As a user when I send request to Google, I want to see 200 Ok status code.

Step 1. Send request [www.google.com](http://www.google.com)

Step 2. Get response

Step 3. Validate if the status report 200 ok

The tests we did in postman using the [www.google.com](http://www.google.com)

* Status code
* Response time
* String match

<https://maps.googleapis.com/maps/api/place/findplacefromtext/output?parameters>

Google API ask for 3 things, need to be provided in order to function:

1. Required Parameter make you to get the exact information and the following information is required that API response:
   1. key
   2. radius
   3. location

* <https://maps.googleapis.com> : Until this part, it is called URL which is the website address.
* Resource: [/maps/api/place/findplacefromtext/output ?](https://maps.googleapis.com/maps/api/place/findplacefromtext/output?parameters)
* Parameter: parameters, after question mark the parameter starts
* Get: Means we send some information to get answer
* Post: means
* Put: Edit API
* Delete: Delete the API location

---------------------------------------------------------------------------------------------------------------------

Date: 27th December 2018

Subject: Interview preparation

Instructor: Guljannet:

* We need to start the interview questions, record your voice and do practice in front of the mirror every day 30 minutes.

Interview steps:

1. You post your resume online, then you will receive some phone calls from the HR or Recruiters.
2. Once they confirm you are qualified

1. Then they schedule the face to face interview. Technical questions are done by the testers. If you pass the technical questions, then the face to face will be scheduled and mostly there are some soft questions by the higher manager and the managing team.

* The higher manager really cares in interview about personality, soft skill and be someone to be trusted by the managing leaders. Try to act away that the higher managers like you. Nice, friendly, while you are answering questions, pay attention and answer based on the answers. You can ask back some questions from the interviewers to make it a conversation. Smart questions will be helpful during the questions too. Be nice, dedicated and someone that you can work with a team. Just make the people comfortable during the interview.
* Before going to the interview, you can read the email and do some research about the interviewers, then prepare your interview answers based on their desires that what they want to hear.
* Communication is one of the important parts that can make you success.
* Behavioral questions are also a chance for us to show how dedicated you are. They ask you about your strength, weaknesses and things like this.
* Story and examples are very important, and it will impress the interviews. They might be experiencing the same challenges and they will decide to hire you.
* All stories must be ended with a happy path and have a positive point.
* Describe a time you worked under pressure?
  + I always work under pressure, such as good pressures are new tasks and good products. I prevent to get stressed. Sometimes, the developers deploy their codes late. So as a good tester, I don’t sit just do nothing. Because, I am writing test cases, Automation plans and test scenarios to better accomplish my jobs.
* What is your strength?
  + My strength is to prioritize my job and do the most important part first.
* Why did you choose software QA engineering?
  + Just don’t talk about the money to answer this question, stay away from the negativity.
  + You can say that you love to be in this environment, I have all the knowledge, communication, I love to think out of box, I always want to do something that other people struggle to do it and it makes me feel good.
* Talk about your application?
* Tell me about yourself?
  + This is important to practice a lot and summarize your answer in a couple of minutes to use the keywords.
* Describe your day to day or typical activities?
  + I come morning, check my email to make sure if there is any meeting, prepare for the daily scrum meeting, (knowledge transfer meeting -> Any functionality or tools is developed you can learn from the teammate or you can teach the teammate, you explain, or they explain how the application works). Review the requirements, working on the opened Bugs or tickets, communicate with the developers, taking part in Bug Scrum Meeting (If you find a Bug, then discuss it with the developers to find a solution for it), it is also a part of tester’s job to attend the meeting if the Bug related to your testcase.

* + Bug Scrum Meeting: It is usually 30 minutes and happens around 15:00 and the tester who is assigned for that certain Bug, will attend the meeting. If there is no Bug, then the Bug Scrum Meeting will be canceled.
* Are you a team player or lonely wolf?
  + It is better to be a flexible, working with team and along. I also preferred to be cross-functional that can fill our partner’s position if unexpected situation come up. It is better to be flexible, I can handle both situations. I can work alone or work in big team. Scenario (Maybe you work in a team, they have an already developed application and you just work in maintenance part. They just need 1 tester, if you say that you prefer to work a team, then you may lose the job opportunity).
* What do you prefer to work with big team or small team?
  + The correct answer will be that you are willing to work with either big team or small team and both have pros and cons. Small teams are less effective.
* Why did you apply to this QA position?
  + Talk about the team environment, in order to answer this question, you have to know the job description and do some research on the company before going to the interview. You have to know the mission statements, values or team cultures.
* Why you are looking for this job?
* What is your strength and weakness?
  + Weakness: Too critical in my work, very detail oriented and sometimes cause me problem. If I provide information about a Bug to the developer, I provide a lot of details that sometimes it is extra and no need to provide that much details.
  + Strength: I am eager to learn easily things and I am always thinking out of the box.
* The interviewers will ask at the end of the interview? Do you have any question?
  + You can ask about the following issues:
    - What is your framework?
    - What is your team structure?
    - Are you challenging during your sprint?

Homework:

* Prepare 3 best stories related to greatest achievement, biggest challenge, strength

Date: January 3, 2019

Teacher : Nadir

* USE OF JIRA:

1. Tell me about yourself?

* Experience (started as a manual tester / moved to automation)
* Industry (health, IT, Retail, Banking, Insurance)
* Platform (windows, IOS, Linus)
* Application (web-based application, client server applications)
* Languages (Java, SQL, JavaScript, Python)
* Testing tools (Selenium, TestNG, Junit, Cucumber)
* Building project (Maven, Ant)
* IDEs (Eclipse, Intellije)
* Project/Bug tracking tool (Jira Xray, ALM)
* Continuous integration (Jenkins)
* Repository (git/svn)
* Framework (pom framework, hybrid fframwork)
* Methodology (Agile, Waterfall)
* Test types (smoke tests, regression test, functional test)
* Quality (a team member, easy going, detail oriented, open to learn new things)

Date: January 17, 2019

1. How to prepare / study from this point?
2. Tell me about yourself?

* First of all I would like to thank you guys for giving me the time and opportunity.
* I have been in the IT industry for 6 years so far and I am currently an S**DET** on my team. After working in different types of Industry I have gained domain knowledge in networking , banking, and healthcare.
* I started off as a manual tester and eventually became proficient in automation testing, I have been exposed and have experience in Frontend testing and backend testing. The tools I used in my current project are Selenium WebDriver, Java, TestNG , Cucumber for front end.
* SQL, JDBC, and oracle database for back end. I have also worked on API testing in my project and I used Restful , postman, and restassured library
* I also used GitHub for version control, and Jenkins for continuous integration/deployment, and maven for framework organization and  Jira for project management and I have experience with Amazon AWS cloud machine server management.
* I have been exposed to waterfall methodology early in my career and currently I am in an agile scrum team and participate in meetings and analyze artifacts
* I have also constructed and design my own framework that are data driven and behavior driven based on POM and I am also able to adapt to any existing frameworks.
* I am a positive person and I love helping companies achieve their goals and grow with them . I’m a team player and my current team would say that I am reliable and have good communication skills to ensure that we meet our goals during the sprint.

1. Tell me about your day to day activities?
   * + - I schedule my Jenkins to run the smoke tests at 6am. It runs for about 30 minutes, so by the time I get to the office, the results are ready. If there's an issue, right away I'll inform the rest of the teams via email or slack.
       - Then, as soon as I'm sure there's no fires to put out from the smoke test, I get some coffee and really start my day.
       - I'll check my emails and make sure nothing came through after I left that requires immediate attention.<NS1>Also if there's any messages from yesterday afternoon that I hadn't replied to, I'll take a few minutes then to clear that off my plate.
       - I'll give the Jira board a once over, see where things stand. If there's something that's almost done I'm making note of it so we can try and get that completed and move it to the Done column.</NS1>
       - Then I start getting ready for our daily stand-up. As always, we'll cover what we each did yesterday, then discuss what we're going to do that day, and if there's any impediment we've come up against, or any we see coming down the road.<NS2>We usually update the Jira(<-\*?Phrasing?\*) during the meeting(\*?\*)We're religiuos about time limits. If there's anything we need to cover that won't fit into that 15min box, we schedule an off-site.</NS2>
       - After that we break and dive in to it. I'll head to my desk, unless it's the very first day of the sprint, then I'll take part in Sprint planning meeting, where I'll get eyes on the User-Story that we'll be working.
       - <NS3> In my team right now, we have 3 developers and 2 testers. We tell them our points, they tell us theirs, and come to a consents as a team, justify the pointing with each other to make sure we're clear on how much time we'll need in total. Of course, it'll take a little time for the Dev team (<-\*?WC?\*) to complete the User-Story, but I'll normally go ahead and start on High-Level scenarios(<-\*?DEF?\*). I have a great relationship with the developers on my current team, so I know now what IDs I can expect and can even start fleshing out the test script(\*WC\*) to some degree, especially if there's some standard functionality in there, log-ins, etc..(<-\*?IS THIS CORRECT?\*) </NS3>
       - Also, if the manual testers are busy and I need to pitch in there, I will. Otherwise, I'm maintaining my framework. I'll be making sure the test environment is ready, that the framework, and the page object models are in place and ready to go.
       - If its toward the middle of the sprint, then our Test Scenarios are done. We'll go over them with the Product Owner, the BA, and the Dev team and make sure they address all the relevant acceptance criteria and are absolutely ready. At this point I'm focused on writing and executing the test scripts. I'll run them manually first, then finalize the scripts and begin automating them.
       - If we're in the end of the sprint, I'm readying whatever demos we have to make sure they're all set for release.
       - We have a Retro Meeting every sprint . We take a few hours to focus on what went well over the past two weeks, what didn't go so well, and how we can take these lessons into the next sprint. I really appreciate these aspects of Agile, these ceremonies. They're all basically about sharing knowledge, whether in little bites at the daily scrum, or in big meals at the end of sprints, when we get together for the Retro , or code review to learn from each other and clear out any spaghetti that might have built up over the past Sprint.
       - My favorite part of day-to-day life in my current project are our brownBag meetings. Every week we have an hour where we just share our knowledge with each other. I'll help them build their Java and automation skills, or someone else might share another language they're learning -- it's been something I've been taking with me in my career to all my projects and it helps to keep us all learning, building our skills and opening new opportunities.

------------------------

Hi, everyone,

the parts between the <NS></NS> tags is stuff I made up based on things I've heard in conversation or class or read somewhere, so take double check it. The rest came from the source document, I just changed the tense a bit and fluffed it up some.

You can all come up with better versions of this. And when you do please send me a copy.

Thanks, Albert

1. Tell me about your responsibilities?
   * + - My number one responsibility Smoke Test
       - I am responsible for regression test, before the release I need to make sure the application is tested from a regression prospective.
       - I am responsible to take part in daily stand up meeting.
       - I participate in test planning, as a tester I participate in sprint demo and I handle the demo as a tester to show them all functionalities.
2. Tell me about your project?

---------------------------------------------------------------------------------------------------------------

Date: January 24th, 2019

Instructor: Guljannat

* + - * What is Software Development Life Cycle?

1. Requirement gathering: Requirement is the initial step in SDLC and you gather it to develop an application.
2. Design: Design takes places in the second stage by the software architecture engineers. The design phase comes after a good understanding of customers’ requirements, this phase defines the elements of a system, the components, the security level, modules, architecture and the different interfaces and type of data that goes through the system.
3. Implementation or coding stage: Developers are developing the modules (DEV Environment), Unit testing happens by developers, they also deploy it to Quality Control (QC). (when the bottom of QC is active, then the testers start their project or work period, they usually say they are working in certain tickets like; login ticket, which is the term used by testers).
4. Testing: It happens by the testers, (LOE level of efforts). Always put extra hours for testing to divide how to implement your testing.
5. Testers will test with different scenario in test environment.
   * + - What is the Software Testing Life Cycle (STLC)?

* STLC is a process followed by software industries to test the software products.
* STLC process is followed strictly to ensure quality products are delivered to the customers within the planned schedule and cost.
  + Requirement Analysis: The previous gathered requirements will be analyzed and also make sure what is testable or identify what will be tested. Scope definition, Create RTM, Automation Feasibility.
  + Testing Planning: Test Plan / Strategy, Resource, Estimation/Plan. As a tester we don’t have to create a test planning, the project manager is creating the test planning. The resource, budget, schedule is also planned by managers for the projects finalized the test plan. Automation tools are researched and selected (if automation is planned).
    - Test Plan: is usually done by the project manager, a document describing the scope, approach, resources and schedule intended test activities.
  + Test development Phase: Test Environment, Create Test Cases, Test Data, Test Scripts, Review test cases (peer review), create test data (if test environment is available).
  + Test Execution: Functional Testing, Smoke, System, Integration
  + Test Closure: Test Closure, Report, Test Metrics.
  + Retesting is to make sure that the previous found Bug is fixed and you retest.
    - * What is requirement document?
* Expectations of users, describing of expectation is a requirement.
* The process to gather the software requirements from client, analyze and document them
  + - * What is module? It is an dependent code that could have one or several functionalities inside.
      * What is build? It is a combination of the 2 or more modules.
      * What is the build deployment? Whenever a built code is pushed to the lower environment (testing environment), it is called build deployment.
      * What is the difference between build deployment and release?
        + Deployment is lower environment and release happens at the end of the project where an application is completely developed, tested and ready to be released.
* What is a peer review? It is reviewing or checking each other’s test cases. After creating test cases, as a tester, we review our each other’s test cases and it is peer-review.
* Who approves test cases? Whoever is doing the peer review, then he/she can approve or disapprove the test cases.
* How can you tell when enough test cases have been created to test a system/application/module?
  + Requirement Traceability Matrix (RTM): At the end they want to know which task cases covered and which task cases are not covered. Usually created in excel sheet.
* Test Planning: It is done by the team lead and it is an detailed word type of document which explains all the project test details.
* Test Strategy: As a tester we don’t have to create a test planning, the project manager is creating the test planning. The resource, budget, schedule is also planned by managers for the projects finalized the test plan.
* Test Case: It is created by the testers and it is a detailed step by step
  + - Test Plan: is usually done by the project manager, a document describing the scope, approach, resources and schedule intended test activities.
* What is entry criteria and exit criteria?
* Do the test plans also change along with requirements? If the requirement changes, it could affect test cases, test scenarios, design and almost everything.

=====================================================================

Resume:

- It is paper version journey of your life,

- You can't make a standard Resume, you need to make your resume based on your skills and experience.

- What is a Resume all about?

-> All resumes are being red by the recruiters, the way it is setup

-> Can you talk the talk? Can you backup you "talk"?

- PHP, C#, any database that runs in AWS. It is a massive itnerest, most of the databases are connected through these important programs.

- Sometimes if you have a skill to decrease the laod of the work by increasing the number of servers. Usually, it happens once or twice a week.

- Show you can "walk-the-walk"?

-> You need to take responsiblity and you have to have the patient to be able to do it. Recommended that you have to have multiple Resumes, you can make a LinkedUp and you can upload different Resumes, the different Resumes are for different purposes, if you upload your Resume, it doesn't mean that you are looking for a new job, but you can just have it in case that something happen, you should have some alterantive ways to navigate.

-> Constantly change and update your Resume, you can upload it using the linkedin and other social pages.

-> Don't make a stuck of pages for the Resumes, the good resume should be between 1 - 3 pages. The recruiters don't have time to read multiple pages of the Resumes. A 30 pages Resume, called C.V (Curriculame Vitae). If you work in Europe, they want to know from the day taht you went to school up to now and that could be 20 to 35 pages. But, here in USA, we don't have to put that much inclusive information.

-> You have to highlight some important programs that you are interested, give some details and put some additional information to show what you did and any project and make it highlighted.

-> Specifically, what do you love and what do you want to do, put in your Resume.

-> When you come to the interview:

1): Extremely knowledgeable, but one mistake made him to fail, he said that he wnated to become a director in the next 2 years. He was insisting that he wants to run the team, he didn't talk about the position that was he applied for.

2): Gentlman came, he talked about right topics, he explained about sprints and scrum masters and he pulled out 11 pages of the AWS that he has done and understood AWS cloud and he could back up his talk. 250 K was suggested for him for his 5 years experience.

3): The third person Resume looked good, but he could not denfend from his Resume, His resume was saying a lot good things, but he dressed preffional and respected the audiance. First impression is very important, be polite, set after the interviers sit.

6 hours of interview, the scope will be:

- Cultural fit,

- personality fit,

- project fit,

85 % percent, it is important that you cultural fit, you need to communicate with the people and it is mandatory.

-> If you can't communicate, it will be a problem, the environment is pure Agile, you have daily Stand-up, every single day 15 minutes, what did you do yesterday, what do you do today, any blocker or any issue came up.

-> How about selling yourself?

- Can you back up your talk? it is related on how to sell yourself, It is a waste of space, the objective is the job that you have applied for. You have time to explain what did you do in job, what did you do at school and don't talk about the extra unnecessity stuff.

-> I don't care about all Technology, what did you do with the technology, what you say, can you backup and it is super important.

-> You have to able to explian what yo did.

Resume:

There are 3 sorts of Resumes:

1) Infographic Resume:

It is getting you to get the job very easily.

Proffessional Profile:

- Critical Thinker

Selected Accomplishment:

- put details about important projects that you worked. The budget and everything which is important should be highlited.

- Junior Developers: You have to talk about the projects, classes you took in the school, if you wrote even small projects, you have to explain that you have done something. Nobody cares what you studied, it is important that you explain what did you do with the technology.

- The group projects are very important, you have to exlain what did you do working as a group and team member.

- Invissible drill, watch it in Youtube, at the end of the video, perception, they tell you about situation awareness and it is a great video.

-> You have to explain that what did you do and what was the outcome, it should be in your Resume with the long details.

-> PHP, UI, React, C++, AWS cloud are very important to learn in the next 5 years

2) Classic Resume:

No special fonts, it is pretty clear, a summary, work history, education, plese make it less than 3 pages. Nothing wrong with it and it is usual.

3) One-Page (Single Page Resume): The reason the instructor built was, he worked on a firing Range, they did not care about the Resume, It was just giving to them, you can use this Resume as a back-up document, don't make it one-page and it can be lost. It is good to have one page, if someone ask about the summary of your resume.

-> You have to look for the MyMaps, it let you lay down everything that you have done and you can take it from my map and through it to your Resume.

-> What is your weakness? I always listen to the people, but I have a solution about it. I am putting the dots together and once I have a complete image, then explain it. sometimes it make poeple upset.

-> What is your greatest strenght? Why we can not do it? this is a good perception to have and it is important to think like why can't you buy the company or other thing.

-> Whatever you make everyyear, make your 41ok increased, put another 10 % on the top.

-> Try to get promoted in your company,

-> Body language, eye contact is important, you are very careful and not shacking your legs. Put suit and don't wear blue jeans. Act proffisional and you will take the attention of the interviers, use the white board if there is one in the interview room.

-> You need to communicate appropriately to be hired.

-> At the end of the day life should ask us "Are you sure you want to save the changes?"

- You need to have a good profile, a good background and you need to stay focused on that.

-> Which Resume is the Best? the Resume should be 2 - 3 pages, even you go with zero experience, you have to explain the project you worked.

-> Bullet points, should be 2 lines, not more than 2 lines. It should be well written from English language and technical prospective.