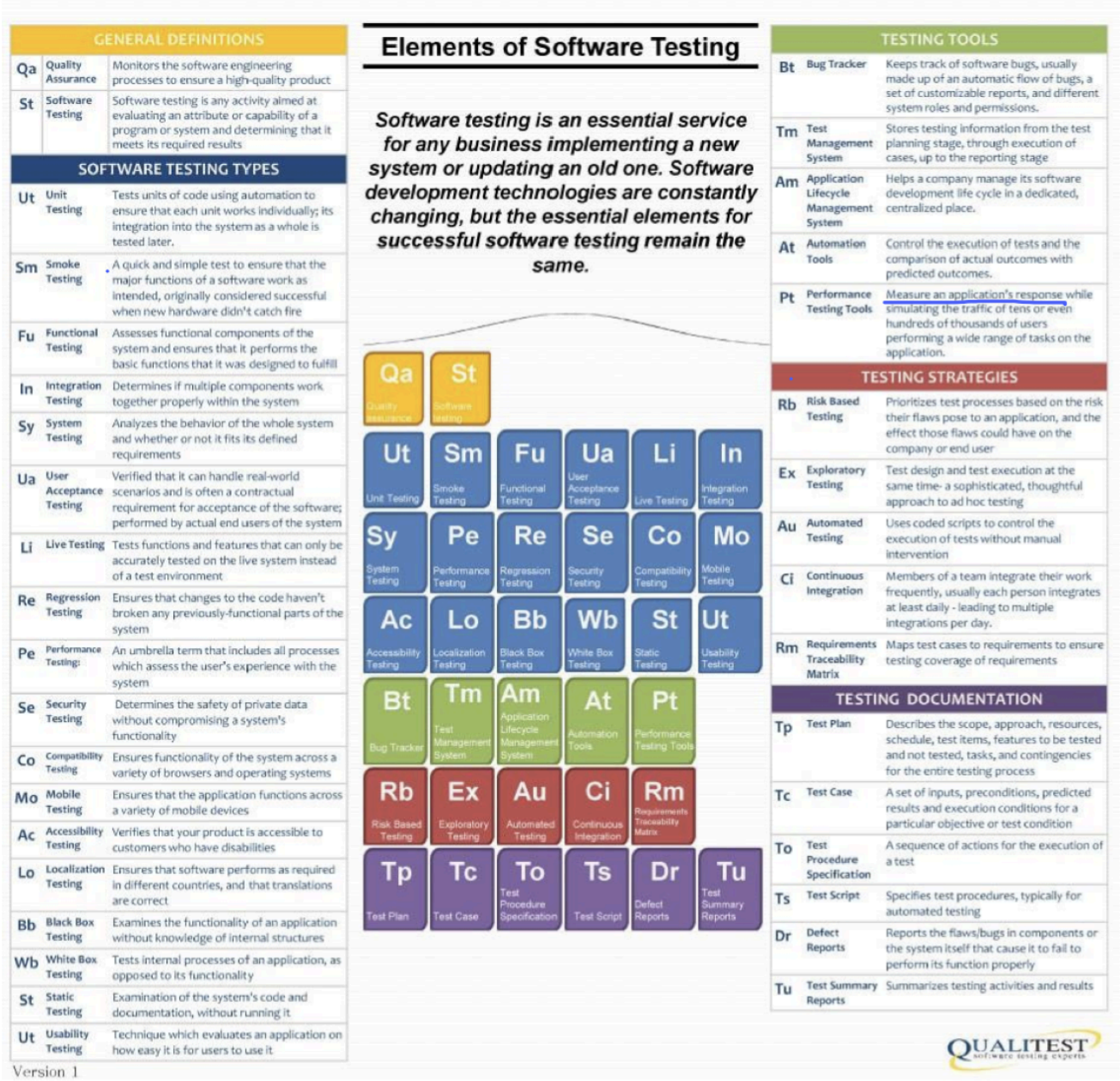


Entry Gate

- Ưu tiên ứng viên đã có kiến thức cơ bản về Automation Testing hoặc đã sử dụng/tìm hiểu về một trong các Framework: Selenium, Appium, TestNG, Cucumber…
- Có kiến thức cơ bản về Java Core/ SQL Queries, OOP hoặc Am hiểu về vòng đời sản phẩm phần mềm và các quy trình kiểm thử (Manual Testing).
- Kỹ năng giải quyết vấn đề và troubleshooting
- Có khả năng làm việc độc lập và làm việc theo nhóm

Recommend from Experts

Ministry Testing, HelloTalk, Livestream in Youtube

1. About 3 months training for fresher

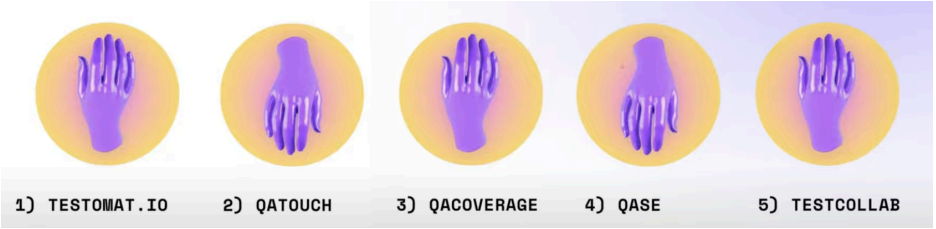
		Exercise
Fundamental of Software Testing	56 pages of theory	Testing my Understand
SQL Query and Java	Java in 15 mins SQL Commands	
API Testing	Postman FastAPI for image classification	
Effective Test Case Writing	Bug Requirement	Web Automation Project with Selenium
Performance Testing (only for website have enormous user like Shoppee)	DNS Jmeter	
Automotive Testing	Everything	

2. Tool

2.1 Bug Tracker

- Top 5 Best Tools in 2024: Monday, Bug Herb, Zoho Bug Tracker, Click Up, Teamgantt

2.2 Test Management System



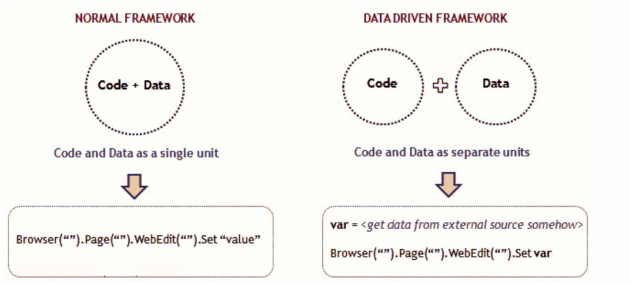
2.3 Applied Life Cycle Management System

2.4 Automation Tool

2.5 Performance Testing Tool

- Top 3 Best Tools in 2024: Jmeter
3. JIRA Integration

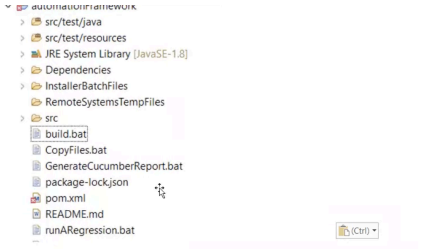
4. The Lifecycle of an HTTP Request
- Understand how a request sent to the server works. A life cycle of a request.
- Code: 15 MINS
5. How can we know which test cases get automated?
6. Data Driven Testing Framework
    - Can execute the same script with multiple sets of test data.
    - Re-usability of code
    - Improves test coverage
    - Faster Execution
    - Less maintenance
    - Permits better error handling



7. Basic of Selenium
- 10 mins explain

HOW TO EXPLAIN SELENIUM AUTOMATION FRAMEWORK

1. **Programming language:** Java, Python, .Net
2. **Framework Type:** Data driven framework, keyword driven by using Page Object Model
3. **POM:** As per the Page Object Model, we have maintained a class for every web page. Each web page has a separate class and that class holds the functionality and members of that web page. Separate classes for every individual test.
4. **Packages:** We have separate packages for Pages and Tests. All the web page related classes come under the Pages package and all the tests related classes come under Tests package.
   For example, Payment Page and Login Page have separate classes to store element locators.
5. **Properties file:** This file (config.properties) stores the information that remains static throughout the framework such as browser-specific information, application URL, development, testing URL screenshots path, security questions and answers, username, passw(Ctrl)·ing , development, testing URL, etc.



The above screenshot illustrates a standardized maven project. As per the above maven project, all the tests are kept in the 'src/test/java'and remaining files (such as config.properties, element locators (POM classes), utility files, test data, etc.,) kept under 'src/main/java'.

5.TestNG: Using TestNG for Assertions, Grouping, and Parallel execution.

6. **Screenshots:** Screenshots will be captured and stored in a separate folder and also the screenshots of failed test cases will be added to the extent reports.
7. **Maven:** Using Maven for build, execution, and dependency purpose. Integrating the TestNG dependency in the POM.xml file and running this POM.xml file using Jenkins.

8.Version Control Tool: We use Git as a repository to store our test scripts.

9. Jenkins: By using Jenkins CI (Continuous Integration) Tool, we execute test cases on a daily basis and also for nightly execution based on the schedule. Test Results will be sent to the peers using Jenkins.

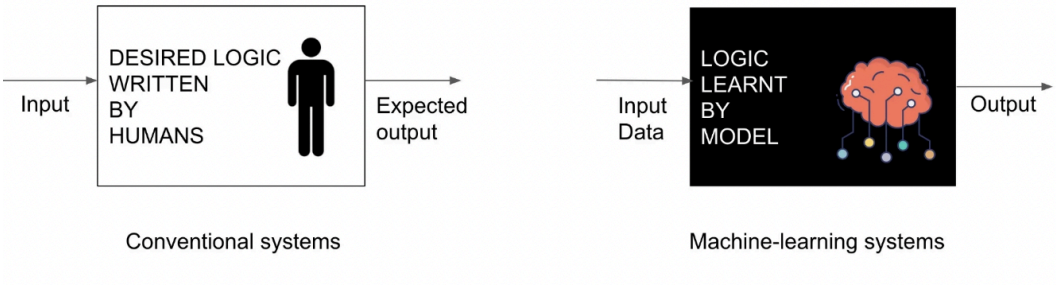
10. Installer Batch Files: In case of desktop-based applications installation you would be keeping PowerShell batch files.

11. src/test/resources: AutoIT Scripts, Sikuli images, licenses to make your application under test up & running, etc files kept

12. Test Base Class: Test Base class (TestBase.java) deals with all the common functions used by all the pages. This class is responsible for loading the configurations from properties files, Initializing the WebDriver, Implicit Waits, Extent Reports.

13. Reports:

8. How I contributed as a Tester to a Machine Learning System: Opportunities, Challenges and Learnings



The model identifies these patterns and predicts the probability of future events.

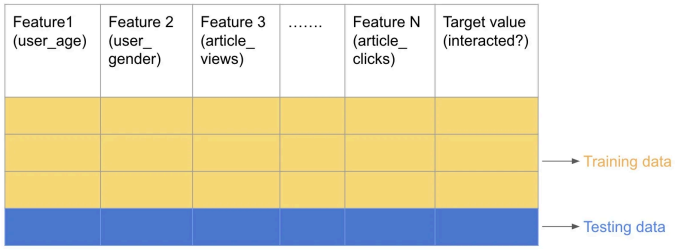
For example, our project was to provide article recommendations for our users. So, a huge amount of data like article interactions, users characteristics and user behavior on the platform was collected and fed to the model. The model then learned patterns in the data and formed rules to predict future possibilities of users interacting with the article.

The process is split in two phases:

Phase A - Learning of model where model learns and identifies the patterns from data and comes up with logic.
Phase B - Deployment of model where new unseen data is supplied to model. The model applies the learned logic to the new data and makes predictions.

Step 1- Data collection: I would indeed say, "The higher the quality of data, the better the training". So, as tester we can help developers check/fix all quality dimensions like: Missing values, Incorrect values/data types, Inconsistencies, Duplicates, Security issues, Business rules, Biases, Outlier data, and so on

Step 2- Building of model features



As a tester, I helped in reviewing all features, standardizing rules applied to them, and applying product knowledge to find bugs/enhance features. Some examples: While calculating the feature of "views on article", unique user views were counted, meaning if a certain user sees an article multiple times, then it was only counted as one view.

Step 3 - Training and validation of model:

After the data is collected and features are decided, it's time to feed this data into the model and train it.

whenever there was a change in rule of filter or re-rankers, I tested the relevant change and sometimes even found crucial bugs.

It's definitely crucial to test before deploying whenever any changes are made in the setup.

9. Skills of a Tester

- **Time Management**
- **Critical Analytical Thinking**
- **Clarity and Practical Logic in Test Cases**
- **Organising Knowledge Documentation**
- **Efficient Communication**

+ Search and be able to express well what you find.
+ Good relationship in your team.

- **Troubleshooting**

+ Ask the question to the correct person.
+ There is no bug-free product.
+ Take ownership and be patient

10. Question to ask after interview

- Det what the company values

  What do you value the most in your employees?

  Time, Be late,

11. My motivation

- Professional Development: My career goal in 2 years.
+ During the training: Within 5 to 6 months, I start add value in distributed team meeting (Asking good question, bringing up things that other maybe didn't consider
    + Able to learn software independently (understand the code)
    + Familiarity with client-side test automation tools,
    + Able to differentiate between release testing, smoke testing, sanity checks, and regression testing – and make recommendations to improve quality, and so on
    + Pass all the assignment within the program, then join the real project as offical tester.

Certifications such as CEH (Certified Ethical Hacker), OSCP (Offensive Security Certified Professional), CREST CRT or similar are a plus but not required if you can demonstrate your technical abilities

+ After 2 years, I can professionally do the work as a junior Automation Tester.
+ 5 years is a little too much.
- Full-time position with financial benefit.
+ I can't say I'm passionate about the IT Industry bcs I have followed the academic route during my bachelor's degree. I have another intrinsic motivation, that's my family. Both of my parents are they are already at retirement ages and my younger brother is 14 yrs old which me and parent need to start saving for his Bs.c study. Before I found that education can be a key to satisfy my curiosity and escape of poorness, and I was able to make money by teaching. However now time has changes, I need a stable income for the support of my family

12. How do I learn a topics with Applied Physics (my major)

- Identify the scope, suitable study plan, available online resource
+ Course in University. Listen to have mindmap notes key ideas of the course
+ Discuss with classmate on the subject such as tips, scope ….
- Understanding by basic concept piece by piece by reading and exercising
+ Do the assignment and reach to senior or professor for the difficulty
- Work on real-world project for evaluating the value of information or ideas
+ Search for available seminar, conference about the topics relevant to my work
+ Have myself a supervisor, or mentorx.
- Add value

13. Why do I quit (5 mins)
14. Practicing SQL - 1hrs and Python Interview

I'm aware that Computer Science is a hurricance to me, which I find Software Testing a perfect started for me in IT industry. And with my dedication for learning and the on job-training program, I'm confidence that I can fulfill my responsibility as an fresher in FPT.

15. Test Case for Banking (30 mins)
16. Manual Testing Question (60 mins)
17. 4 Hr practicing Python and other programming skills

- What working as an automation tester is like

  Write a programming scripts to test the functionality or the system

---

1. Applied
2. Self-study and Teamwork
3. Habit in Reading Non-Fiction Book in Vietnamese and English
4. Hobby in Languages to improve my Communic
5. Plan during and after the training

My 20 mins interview

I realized that the developers were well-versed with their domain. To create an impact as a tester, I needed to bring in the perspectives that they missed. We, as testers, are blessed with the great skill set of asking the right questions, understanding the big picture, thinking out-of-the-box, applying deeper product knowledge, challenging the status quo, etc. If these skills are applied to test ML systems, a lot of the issues could be prevented.

Skills	**Quality**
1. Ability to learn software independently 2. Familiarity with Microsoft DevOps for tracking and collaboration 3. Familiarity with client-side test automation tools	1. Proactive about finding bugs that may not be obvious and identifying use cases requiring retest after fixes or feature changes are built 2. Expedient about performing tests according to our weekly release cycle

4. Able to differentiate between release testing, smoke testing, sanity checks, and regression testing – and make recommendations to improve quality 5. Power user level skills in Microsoft 365 6. High level of fluency with English, both spoken and written – including technical terminology 7. Able to write documentation for our support team 8. KN với Python và API Automation Testing, hiểu biết về parallel testing cách set up 1 flow và troubleshot QA process	3. Growth-oriented to help us continue to improve our testing model and release process 4. Encouraging and helpful when interacting with customers as they report bugs 5. Keeping processes as simple as possible to avoid adding unnecessary overhead 6. Expertise – with triaging and documenting bugs, and with writing / tracking test cases 7. Empathetic to users, regardless of which permission / role they have 8. Technically curious to learn our tool thoroughly and enough about related tools to find ways ensure quality

Proven experience in web application security testing, preferably in a remote capacity. Proficiency in using BURP Proxy for security assessments and familiarity with other relevant tools such as ZAP, Nessus, etc. In-depth understanding of web application security concepts, OWASP Top 10 vulnerabilities, and secure coding practices. Strong knowledge of industry-standard frameworks and methodologies like OWASP Testing Guide and PTES. Excellent analytical and problem-solving skills with the ability to think creatively to circumvent security controls. Effective communication skills to articulate complex technical findings to both technical and non-technical stakeholders. Ability to work independently in a remote environment while also collaborating effectively within a distributed team. Certifications such as CEH (Certified Ethical Hacker), OSCP (Offensive Security Certified Professional), CREST CRT or similar are a plus but not required if you can demonstrate your technical abilities	Conduct thorough security assessments of public-facing web applications using BURP Proxy and other relevant tools. Identify and exploit security vulnerabilities including but not limited to SQL Injection, Cross-Site Scripting (XSS), Cross-Site Request Forgery (CSRF), Authentication and Authorization flaws, etc. Perform manual and automated security testing to discover potential weaknesses and gaps in web application security. Collaborate with development and engineering teams to communicate identified vulnerabilities and assist in remediation efforts. Document findings, testing methodologies, and recommendations in clear and concise reports. Stay updated with the latest security trends, vulnerabilities, and best practices in web application security. Adhere to project timelines and deliverables while maintaining high-quality standards in testing and reporting. Provide guidance and mentorship to junior members of the security testing team as needed.

1. What is the Software Development Life Cycle?

- SDLC is a step by step procedure to follow for developing new software, from the requirement to the production. Phase:

+ Requirement Gathering: BA and Product manager will be involve
+ For the Design (Study of technical, resource, financial) and Development: Developer will be involved. They do works like high-level design or low level design
+ Testing: Now the tester will come into picture.
+ Maintenance: Done Then production will start

2. What do I know about STLC?

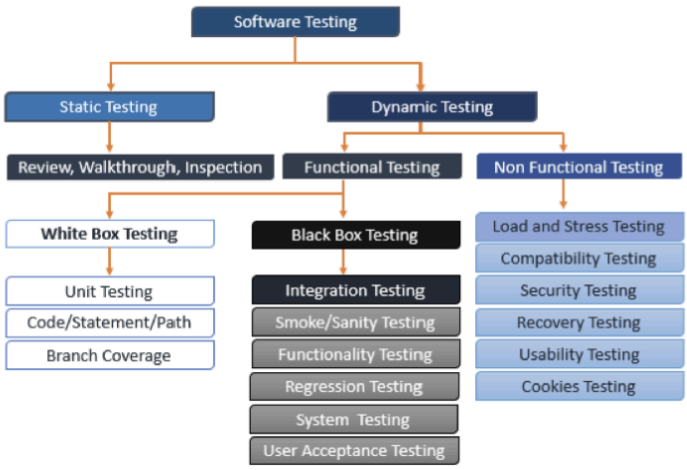
- STLC is about the software testing process, it's a step of SDLC. We will understand the requirements, then we write the test cases and then we will execute it, finally we exist the test and write a defect or bugs report to developer.

3. Models

- Under SDLC, we have: V Model, Spiral Model, Incremental Model, Agile Model

4. VModels, Requirement Gathering, Status Static Testing

- Developer and Tester will go parallel from the requirement to end
  Parallel happened Unit→ Integration → System → Acceptance depend on the corresponding documents, we have SRS doc, design doc, unit test plan, integration test plan.
- During the testing phase as a tester we will do functional testing, but in the initial phase we will review the document, test the documents, talk about how the software will be, what different features should be there. That's come under the static testing, we will verify whether the doc are correctly written or not, completed or not



One the software is ready, we do the dynamic testing

5. Functional Testing

Testing the behavior of a software. How to perform it is here BCS we need input, know s.t about application

+ First the requirement study need to be done (understand the functionality)
+ After that, we start to create the test cases. If the test case required data, then we Prepare the test data

6. Exploratory Testing

- We don't have any document in hand, just learn the application and we will start testing

7. Regression Testing

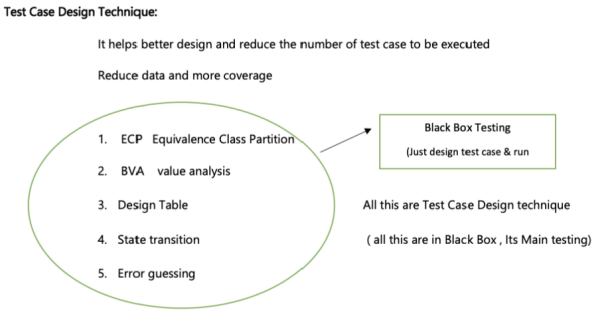
- Done whenever a new change is occurred. We we-running functional and nonfunctional to check if the new code in one model/function doesn't impact another.

8. Writing a test case. (How)

- First: Understand the requirement, the functionality of the software ON the base of test scenarios

9. Requirements
10. Test scenarios (HOW), Functionality

- By Review the FRS and design documents, a more technical format for tester and developer to understand to form the test scenarios.
- Some FRS contains the use cases, which a part of SRS contain what is the input, what is an action, what is outcome.



11. Test Scenario vs Test Case

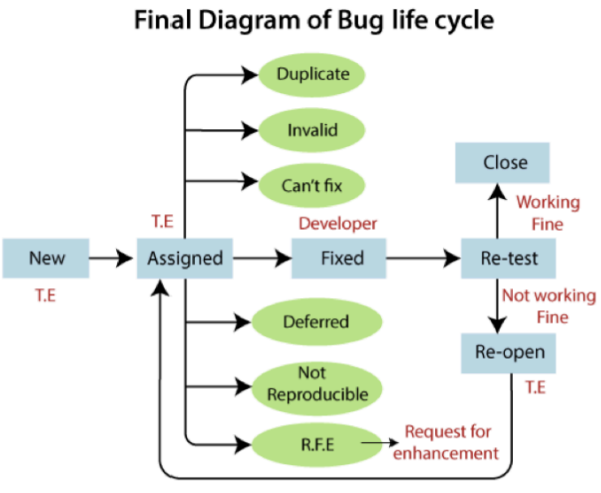
- The test scenarios different to test case: is the detail of it one is what to test, and one is how to test. One scenario can have multiple test case

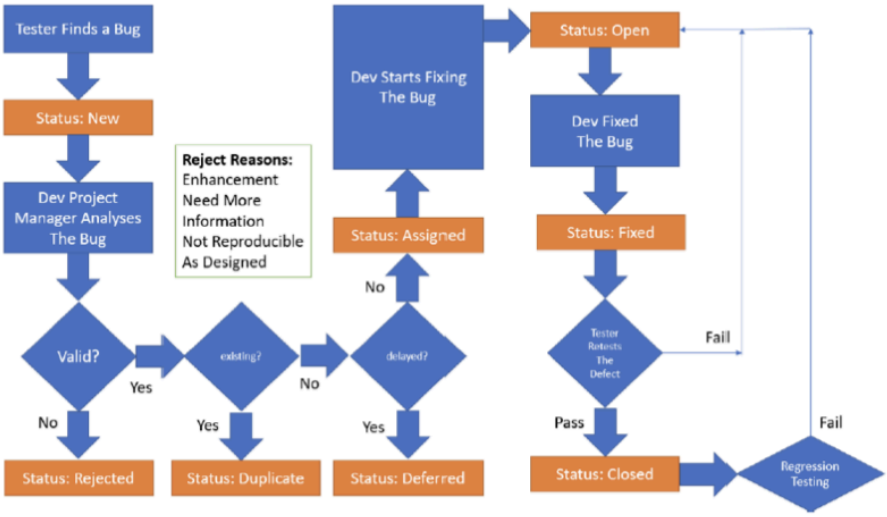
12. Priority vs Severity (what to do?)

- Priority is the urgency of fixing a defector (Hight_immediately, Medium_next release, Low_next version)
- Severity is how much impact it is on the business Or the seriousness of a defector (Blocker_software not process, Critical_main func not working, Major_desirable behavior is display, Minor_ look and feel) which DECIDES by the testing engineer

13. Defect Life Cycle

- After the tester done find bug, status will be new, then dev project manager analyses the bug whether it is Duplicate, Invalid, Deferred, … then the status will be assigned, developer started fixing, status open, then if work fine then close, otherwise they do re-testing, the status is re-open.





14. Agile Process

- ….
- Compare:
+ Adv: Communication is good, and less documentation work. So we can deliver it soon, and good quantity.
  Let them use the dome after we completed some piece of software, parallelly we will use few more features, again we will start development and testing process. (customer no need to wait)

15. Advantages

- Ex: Internet banking applications you want to send money. Test minimum amount, maximum (500tr) per day. How can we test this field:

Use Boundary Values Analysis Technique by check 6 paramarter

+ Check with minimum + 1, and -1
+ Then check with maximum + 1, and - 1
- Ex: One scenario: Upload a file, basically imagine, only accept .PNG, maximum size is 1mb

ECP, divide values into Equivalence Classes ~ Positive and Negative Testing.

P when a software passing the valid input according to the customer requirement

+ One Negative: Verify by upload the doc.file
+ One positive: Verify by upload file with smaller size than 1 mb

N when a software passing the INvalid input NOT according to the customer requirement

16. Object Oriented Programming

- OOPS concept supported by JAVA:
+ Polymorphism
+ Encapsulation
+ Inheritance
+ Class
+ Object

These are data abstraction, these are differences OOP concepts

```
static int plusMethodInt(int x, int y) {
  return x + y;
}

static double plusMethodDouble(double x, double y) {
  return x + y;
}

public static void main(String[] args) {
  int myNum1 = plusMethodInt(8, 5);
  double myNum2 = plusMethodDouble(4.3, 6.26);
  System.out.println("int: " + myNum1);
  System.out.println("double: " + myNum2);
}
```

The different btw overloading, and overriding

+ Overloading is supposed when we create a class in JAVA, we can write multiple methods that have the same name with different parameters. Rule is that the number of parameters should be different. Let say, I create same method with 2 parameters, we can't create it again. But we till created two parameters, we need to **change** the data **type or order of the parameter**.
+ Overriding: only possible with inheritance at least we need to have a two classes, that means whatever method we

created in parent class, the same method if you recreate again in the chain class by **changing the implementation** (not no. parameter, or method)

17. Super Keyword (when)

- Use along with the **inheritance**, supposed in the parent class, you created one method, and in the chain class, you override that method. Now you create an object of the chain class, when you change class, only that method will be called
- But we want to envolve parent class, we use a super cube like parent class variable…

18. Abstract Method (what)

- Method that doesn't have any implementation, only the doc. Can't be as a normal class but can as an interface or abstract class which is create before interface

19. Abstract Class

- JAVA Collection: Arraylist, hashmap
-

20. SQL (5/10)

- `SELECT name`
- `AGGREGATE FUNCTIONS_COUNT/SUM/AVG/MIN/MAX/GROUP BY/ ORDER BY ()`

To give the column a descriptive name, use the `AS` keyword:

```
FROM customers (table)
```

```
WHERE name IS NOT NULL; WHERE name LIKE '%Bob%';WHERE age BETWEEN 45 AND 55; AND OR
```

`GROUP BY`
 	
`HAVING or WHERE`
	`ORDER BY age DESC;`
	
	`OFFSET 10 ROWS;`

```
INNER/LEFT/RIGHT/FULL JOIN orders
```

```
ON customers.customer_id = orders.customer_id;
```

- `CREATE DATABASE dataquestDB;`

```
CREATE TABLE customers (customer_id int name varchar(255),age int);
```

```
CREATE INDEX idx_name ON customers (name)
```

```
CREATE VIEW [Bob Customers] AS SELECT name, agem
```

```
FROM customers
```

```
WHERE name = 'Bob';
```

```
TABLE: DROP(database, table, index), UPDATE(use set and where),
DELETE(row)
```

```
Colum:  ALTER(DROP, ADD)
```

1. Do we need subquery to write max-function

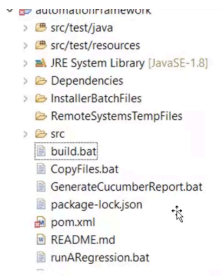
   - Outer query
   - Subquere??

     Single table don't need subquery

1. Basic of Selenium
- 10 mins explain

HOW TO EXPLAIN SELENIUM AUTOMATION FRAMEWORK

1. **Programming language:** Java, Python, .Net
2. **Framework Type:** Data driven framework, keyword driven by using Page Object Model
3. **POM:** As per the Page Object Model, we have maintained a class for every web page. Each web page has a separate class and that class holds the functionality and members of that web page. Separate classes for every individual test.
4. **Packages:** We have separate packages for Pages and Tests. All the web page related classes come under the Pages package and all the tests related classes come under Tests package.
   For example, Payment Page and Login Page have separate classes to store element locators.
5. **Properties file:** This file (config.properties) stores the information that remains static throughout the framework such as browser-specific information, application URL, development, testing URL screenshots path, security questions and answers, username, passwo(Ctrl)ing , development, testing URL, etc.



The above screenshot illustrates a standardized maven project. As per the above maven project, all the tests are kept in the 'src/test/java'and remaining files (such as config.properties, element locators (POM classes), utility files, test data, etc.,) kept under 'src/main/java'.

5.TestNG: Using TestNG for Assertions, Grouping, and Parallel execution.

6. **Screenshots:** Screenshots will be captured and stored in a separate folder and also the screenshots of failed test cases will be added to the extent reports.
7. **Maven:** Using Maven for build, execution, and dependency purpose. Integrating the TestNG dependency in the POM.xml file and running this POM.xml file using Jenkins.

8.Version Control Tool: We use Git as a repository to store our test scripts.

9. Jenkins: By using Jenkins CI (Continuous Integration) Tool, we execute test cases on a daily basis and also for nightly execution based on the schedule. Test Results will be sent to the peers using Jenkins.

10. Installer Batch Files: In case of desktop-based applications installation you would be keeping PowerShell batch files.

11. src/test/resources: AutoIT Scripts, Sikuli images, licenses to make your application under test up & running, etc files kept

12. Test Base Class: Test Base class (TestBase.java) deals with all the common functions used by all the pages. This class is responsible for loading the configurations from properties files, Initializing the WebDriver, Implicit Waits, Extent Reports.

13. **Reports:**