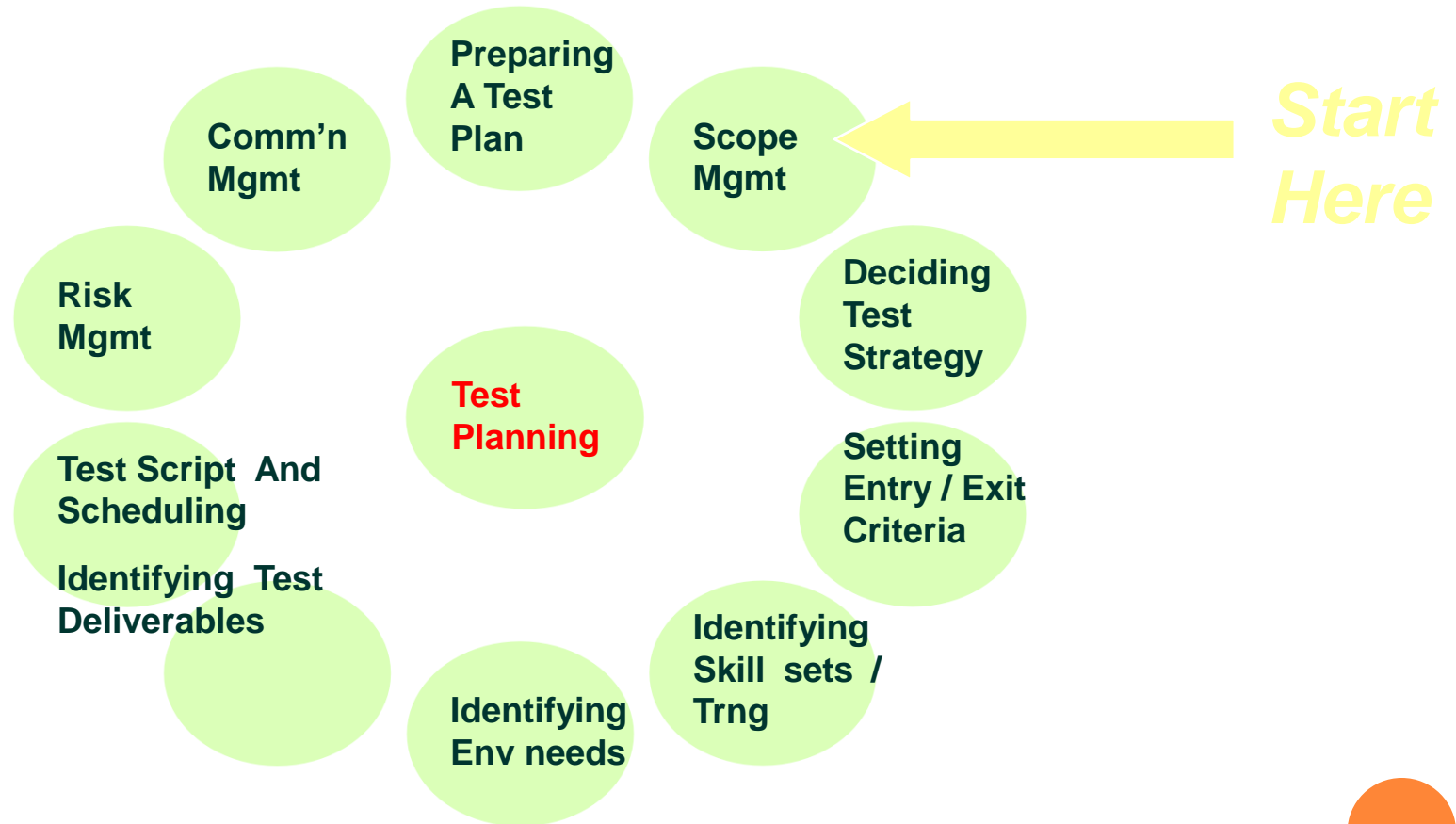
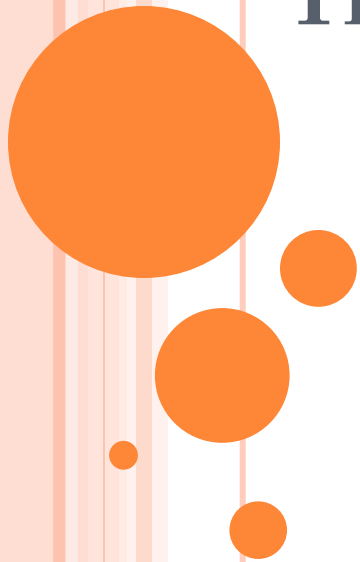




PARTS OF TEST PLANNING



ISTQB CH. 2 - NON FUNCTIONAL TESTS



TESTING HIERARCHY

- Levels of Testing (Unit, Integration, System, Acceptance (Alpha, Beta))
- Types
 - = Functional - Blackbox
 - = Non Functional – Performance, Stress, Usability...
 - = Structural - Whitebox,
 - = Confirmation
- Methods/Modes – Manual / Automated / Semi-Automated



TYPES OF TESTING

- Functional
- Non-functional
- Structural
- Confirmation



FUNCTIONAL TESTING

- Testing based on an analysis of the specification of the functionality of a component or system.

e.g. Black box testing.

- Testing functionality can be done from two perspectives:

1. Requirements-based
2. Business-process-based.

- Requirements-based testing uses a specification of the functional requirements for the system as the basis for designing tests.

- Business-process-based testing uses knowledge of the business processes.

- Business processes describe the scenarios involved in the day-to-day business use of the system.



Project

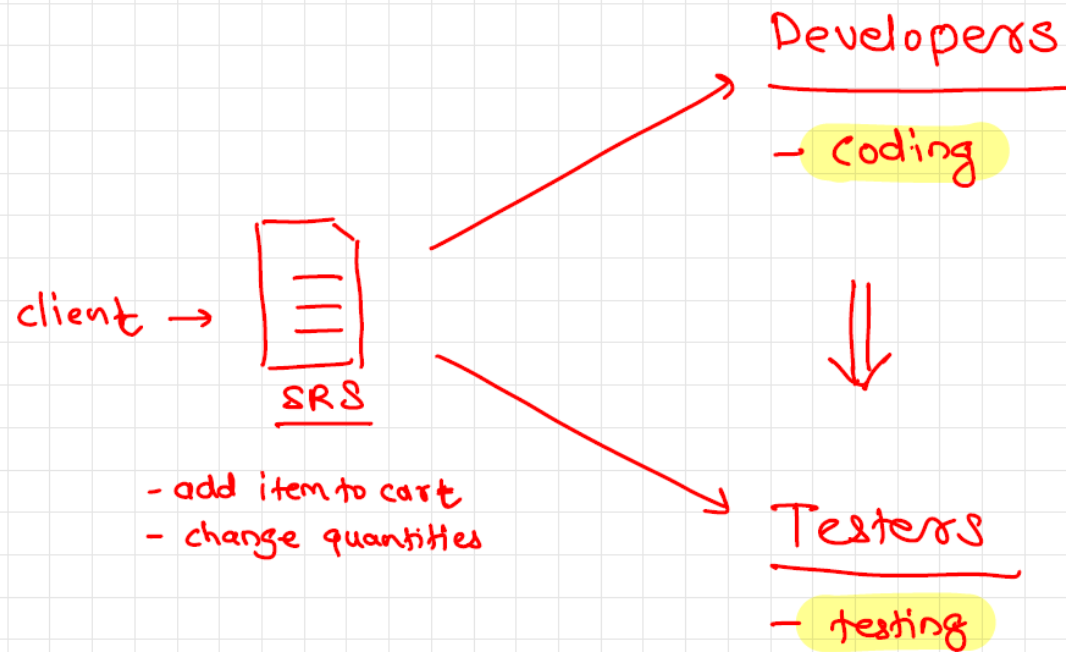
↳ modules

↳ classes

↳ functions

units





NON-FUNCTIONAL TESTING

- Testing the attributes of a component or system that do not relate to functionality,

e.g.

- Usability
- Performance
- Security
- Compatibility
- Maintainability
- Portability
- Reliability
- Efficiency

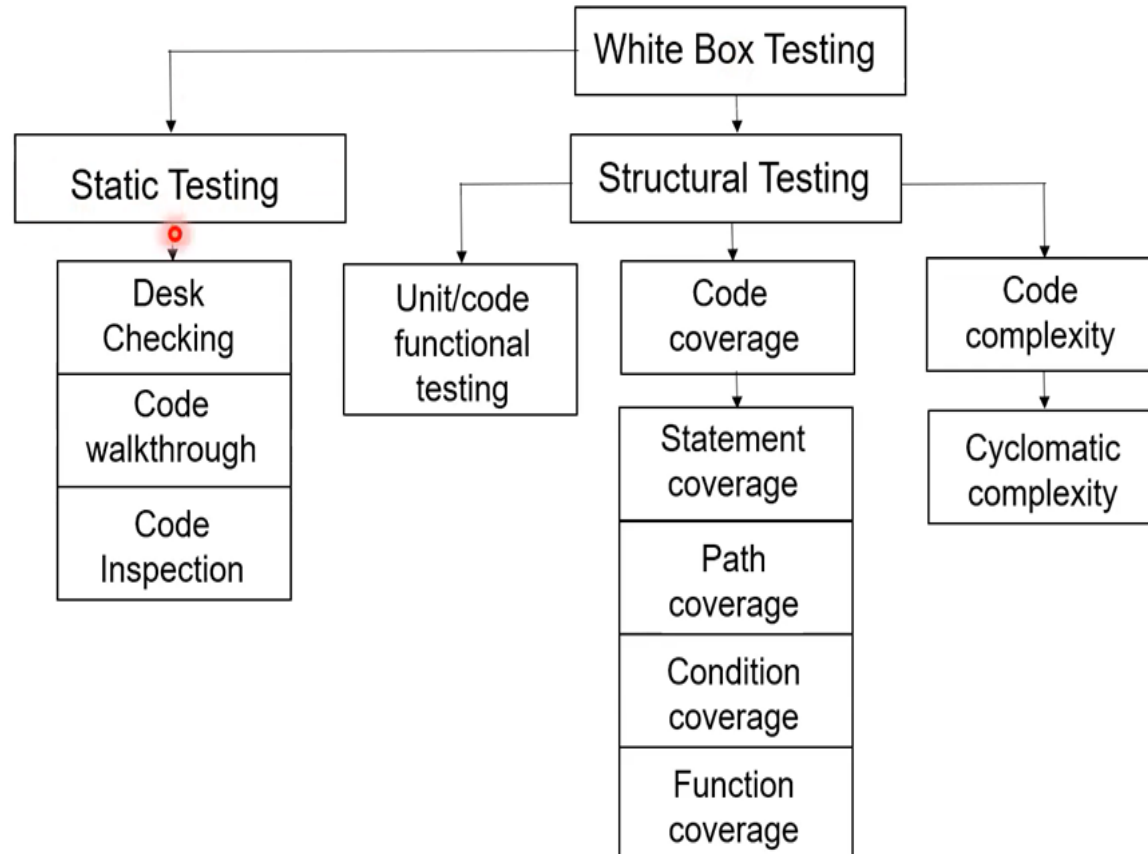


STRUCTURAL TESTING

- Testing based on an analysis of the internal structure of the component or system. e.g. White box testing.

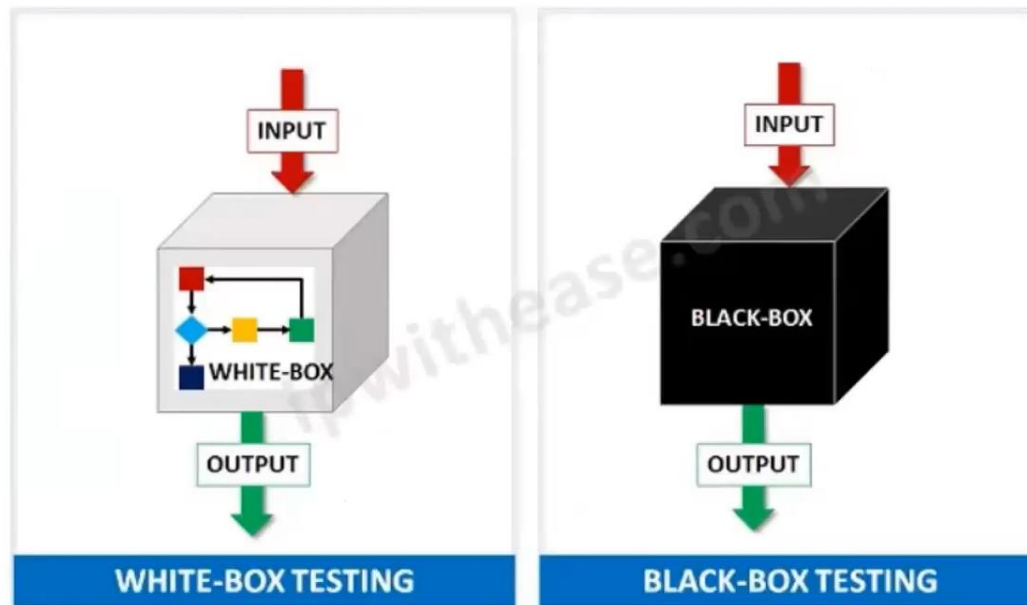


White Box Testing





WHITE-BOX TESTING vs BLACK-BOX TESTING



CONFIRMATION TESTING (RE-TESTING)

- Testing that runs test cases that failed the last time they were run, in order to verify the success of corrective actions.
- While doing confirmation testing, it is important to ensure that the test is executed in exactly the same way as it was the first time, using the same inputs, data and environment.



REGRESSION TESTING

- Like confirmation testing, regression testing involves executing test cases that have been executed before.
- The difference is that, for regression testing, the test cases probably passed the last time they were executed .
- The purpose of regression testing is to verify that modifications in the software or the environment have not caused unintended adverse side effects and that the system still meets its requirements.
- All of the test cases in a regression test suite would be executed every time a new version of software is produced and this makes them ideal candidates for **automation**



MAINTENANCE TESTING

- Testing the changes to an operational system or the impact of a changed environment to an operational system.
- Note that maintenance testing is different from maintainability testing, which defines how easy it is to maintain the system.
- Usually maintenance testing will consist of two parts:
 - Testing the changes
 - Regression tests to show that the rest of the system has not been affected by the maintenance work



MAINTENANCE TESTING

- A major and important activity within maintenance testing is impact analysis.
- During impact analysis, together with stakeholders, a decision is made on what parts of the system may be unintentionally affected and therefore need careful regression testing.
- Risk analysis will help to decide where to focus regression testing.



MAINTAINABILITY TESTING

- The process of testing to determine the maintainability of a software product.
- **Maintainability:** The ease with which a software product can be modified to correct defects, modified to meet new requirements, modified to make future maintenance easier, or adapted to a changed environment.



OPERATIONAL TESTING

- Testing conducted to evaluate a component or system in its operational environment.



USABILITY TESTING

- Testing to determine the extent to which the software product is understood, easy to learn, easy to operate and attractive to the users under specified conditions.



CHECKLIST FOR USABILITY TESTING

- Look and feel : font type, size, colour scheme, background
- Help : On line help
- Standardisation in look and feel
- Progress bar / message for long activity
- Navigations – links, buttons, menus
- Translations
- Visitor feedback , email facility



PERFORMANCE TESTING

- In software engineering, performance testing is performed to determine how a system functions in terms of **responsiveness and stability** under a particular **workload**.
- The process of testing to determine the performance of a software product.
- **Performance:** The degree to which a system or component accomplishes its designated functions within given constraints regarding **processing time and throughput rate**.



PERFORMANCE TESTING

- Load Testing
 - Check the application for normal load
- Stress Testing
 - Check the application for load beyond normal limit
- Endurance Testing
 - Done to determine if the system can sustain the continuous expected load



CONFIGURATION TESTING

- Check response after changing the configuration like
 - Processor
 - RAM
 - Cache
 - Hard Disk Capacity
 - Network Interface Card



RELIABILITY TESTING

- The process of testing to determine the reliability of a software product.
- **Reliability:** The ability of the software product to perform its required functions under stated conditions for a specified period of time, or for a specified number of operations.
- This type of testing incorporates the results from non-functional testing such as stress testing, security testing, network testing, along with functional testing.
- It is a combined metric to define a system's overall reliability. A measure of reliability should be defined by business requirements in the form of service levels.

PORTABILITY TESTING

- The process of testing to determine the portability of a software product.
- **Portability:** The ease with which the software product can be transferred from one hardware or software environment to another.



SECURITY TESTING

- Testing to determine the security of the software product.
- **Security:** Attributes of software products that bear on its ability to prevent unauthorized access, whether accidental or deliberate, to programs and data.



CHECKLIST FOR SECURITY TESTING

- Try to directly access bookmarked web page without login to the system.
- Do not sign-on system, directly try to download the file from the available url, such as enter <http://url/download?name=file> and check if the systems restrict you to download the file.
- Sign out and then click the Back button to access the page accessed before.
- ID / password authentication method: check with valid and invalid passwords, password rules say cannot be less than 6 characters, user id and password cannot be the same etc.



CHECKLIST FOR SECURITY TESTING

- Important information (such as passwords, ID numbers, credit card numbers, etc.) should not get displayed in the input box when typing.
 - They should be all encrypted and in asterix format.
- In the url, enter the following address to check if it can be downloaded restricted files:
<http://url/download.jsp?file=C:\windows\system32\drivers\etc\hosts>, <http://url/download.jsp?file=/etc/passwd>



CHECKLIST FOR SECURITY TESTING

- After session time out try to access restricted page.
- Error messages whether they contain sql statements, sql error messages, as well as web server's absolute path, etc.
- ID / password authentication, the same account on different machines can not log on at the same time. So at a time only one user can login to the system with an user id.
- ID / password authentication methods, entered the wrong password several times and check if the account gets locked.
- Add or modify important information (passwords, ID numbers, credit card number, etc.). Check if it gets reflected immediately or caching the old values.



COMPATIBILITY TESTING

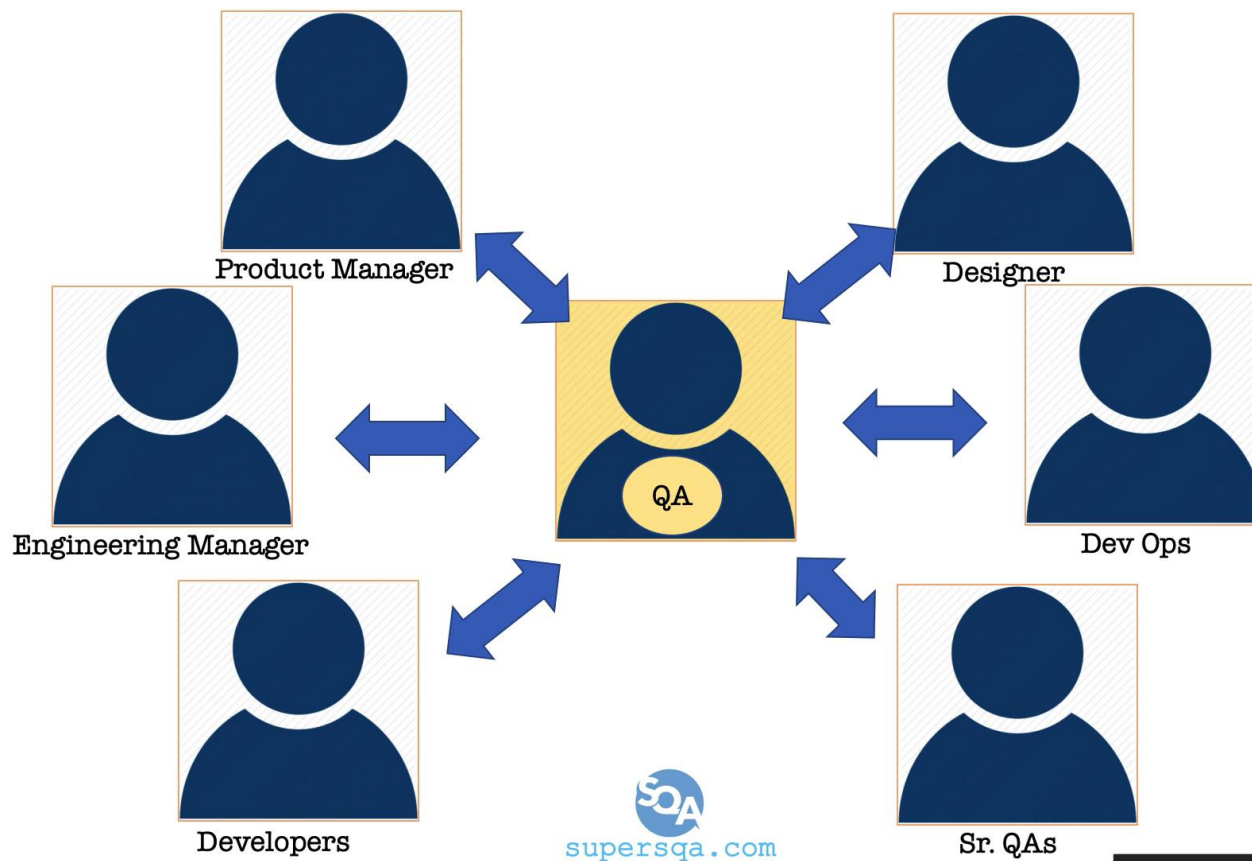
- Required for software products
- Check compatibility with
 - Browser
 - Database
 - OS
 - Hardware



Why is QA/Testing Interesting?

- Software bug can have severe consequences
- Bug = defect
- How about
 - Bug - in hospital software
 - Bug - in weapons
 - Bug - in space ship
 - Bug - in car software
- Much cheaper to invest in finding bugs before release than after the consequence





QA/QE Role

- Testing plus more
- The obvious role is to test the application
- Write test cases
- Document test cases
- Explore/decide on tools
- Think of way to use the product in unintended way



QA/QE Role

- Manage Release - very important role
- Release = Process of pushing code to production/public
- Release is frequent activity
- Application is always changing and we don't even notice
- Example:
 - facebook.com is never down but always changing



QA/QE Role

- Imagine how many people working on a project at the same time
- Imagine what it takes to coordinate
- Code by multiple people can depend on each other or interfere with each other
- QA's responsibility to track and manage changes
- QA's follow strict process to do so



QA/QE Role

- QA's communicate with 3rd party
- QA's coordinate integration testing with 3rd party
- QA's are involved in hiring process
- QA's participate in hackathons
- QA's write production scripts



What is a Test Case?

- Single executable test
- Part/feature that can be tested individually
- One page can have 100s of test cases
- Practice with testing everything around you
- Formally documented
- Information on
 - Steps
 - Data
 - Expected results



Test Case List Example

Case-ID	Title	Discription	Expected Result	Actual Result	PASS/FAIL
1	Valid user login	<ul style="list-style-type: none">- Go to login page- Input email address for registered user- Input correct password- Click on "Log In"- Verify "Welcome" message is displayed	User should be logged in	- User is logged in	PASS
2	Login with wrong Password	<ul style="list-style-type: none">- Go to login page- Input email address for registered user- Input correct password- Click on "Log In"- Verify "Welcome" message is displayed	A red errormessage should be displayed	- Error is displayed	PASS
3	Verify 'Forgot Password' link	<ul style="list-style-type: none">- Go to login page- Click on "Forgot Password" link- Verify a pop-up with field for email opens	Pop-up should open with a field for email and a button	<ul style="list-style-type: none">- No pop-up- Nothing happens when clicked	FAIL



Test Cases Example: Laptop

Test Case No	Test Case	Steps	Expected Result	Actual Result
1	Does it turn on	1. Press the power button	Does the screen come on	
2	Does the OS load	1. Press the power button and wait maximum of 1 minute	Does the login window show up	
3	Does the keyboard work	1. Turn on laptop 2. Type in user 3. Open notepad and type in a paragraph	Verify all the keys you pressed worked.	
4	Does it work both plugged in and with battery	1. Turn on the laptop 2. Unplug the charger or plug it in if it is unplugged	Verify it work both times	
5	Do USB port 1 work.	1. Turn on the laptop 2. Plug a mouse in USB port 1	Verify the mouse works	
6	Do USB port 2 work.	1. Turn on the laptop 2. Plug a mouse in USB port 2	Verify the mouse works	



Test Case Documentation

- Professional tools for documenting test cases
 - TestRail (<https://www.gurock.com/testrail/>)
 - TestLink (<http://testlink.org/>)
- Integration with other tools
- Spread sheet ok but not scalable



What is Manual Tester/Testing?

- Use the product as the customer
- Creates test cases
- Executes test cases manually
- Works with different teams
- Manages release
- Files bugs
- Communicates with overseas teams



What is Manual Tester/Testing?

- Good common sense
- It is an art
- Experience is the only way to get good
- Trainings teach terminology
- Good at breaking things



What is Manual Tester/Testing?

- Positive Testing:
 - testing it works as expected
 - basic functionality
 - what user would do on regular basis
- Negative Testing:
 - trying to break it
 - cause an error
 - error messages



What is Manual Tester/Testing?

- Common terminology
 - Regression test
 - Smoke test
 - Black Box tester
 - White Box tester
 - Gray Box tester
 - Test Case
 - Use Case



What is Manual Tester/Testing?

- Gray Box Testing
- Somewhere in middle of Black and White box testing
- Do manual testing
- No access to the source code
- But know how components of the application work
- Technical but not as technical as White Box testing



What is Manual Tester/Testing?

- Start learning manual testing ASAP to get experience
- I recommend this two sites to start with
 - guru99.com
 - utest.com
- Ofcourse look for courses on supersqa.com (Focused on Atuomation but Manual will be added)



What is Automation Tester/Testing?

- Software Engineer in Test
- Creates test cases
- Writes code to execute the test cases
 - Everything else manual tester does
 - Works with different teams
 - Manages release
 - Files bugs
 - Communicates with overseas teams

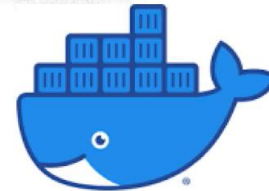
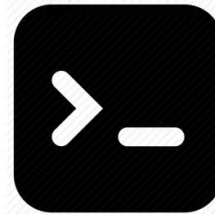


Why Automation?

- Fast:
 - can cover a lot of tests in short time
- Repeatable:
 - so can test multiple times
- Accuracy:
 - not prone for errors



Skills for Automation Engineer



Software Testing Life Cycle (STLC)

Requirements/Design Review



Test Planning



Test Designing



Test Environment Setup



Test Execution



Test Reporting



BUG REPORT

- **Defect ID** - Unique identification number for the defect.
- **Defect Description** - Detailed description of the defect including information about the module in which defect was found.
- **Version** - Version of the application in which defect was found.
- **Steps** - Detailed steps along with screenshots with which the developer can reproduce the defects.
- **Date Raised** - Date when the defect is raised
- **Reference** - where in you Provide reference to the documents like . requirements, design, architecture or may be even screenshots of the error to help understand the defect
- **Detected By** - Name/ID of the tester who raised the defect
- **Status** - Status of the defect, more on this later
- **Fixed by** - Name / ID of the developer who fixed it
- **Date Closed** - Date when the defect is closed
- **Severity** which describes the impact of the defect on the application
- **Priority** which is related to defect fixing urgency. Severity Priority could be High / Medium / Low based on the impact urgency at which the defect should be fixed respectively.

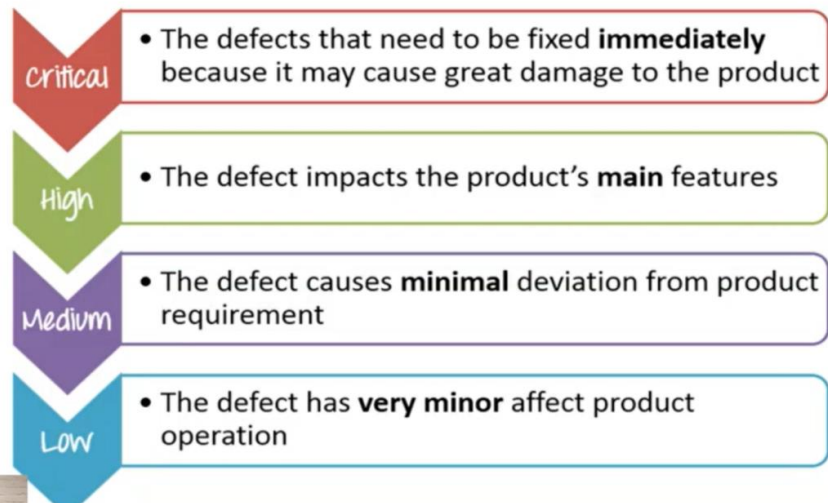
Defect Management Process



Discovery



Categorization



Resolution



Resolution Steps

Assignment: Assigned to a developer or other technician to fix, and changed the status to **Responding**.

Schedule fixing: The developer side take charge in this phase. They will create a schedule to fix these defects, depend on the defect priority.

Fix the defect: While the development team is fixing the defects, the Test Manager tracks the process of fixing defect compare to the above schedule.

Report the resolution: Get a report of the resolution from developers when defects are fixed.



Verification

After the development team **fixed** and **reported** the defect, the testing team **verifies** that the defects are actually resolved.



Closure

Once a defect has been resolved and verified, the defect is changed status as **closed**. If not, you have send a notice to the development to check the defect again.



Reporting

The management board has right to know the defect status. They must understand the defect management process to support you in this project. Therefore, you must report them the current defect situation to get feedback from them.



Defect Metrics

Defect Rejection Ratio

- $(\text{No. of defects rejected} / \text{Total no. of defects raised}) * 100$

Defect Leakage Ratio

- $(\text{Number defect missed} / \text{total defects of software}) * 100$



Bug/Defect LifeCycle

