QA & QC & QE

QA VS QC

pillars of company

p - people --- QC (Testing)
p - process ---- QA
p - product

QA is Quality Assurance.
QC is Quality Control.

QA is process related.
QC is actual testing of the software.

QA focuses on building in quality.
QC focuses on testing for quality.

QA is preventing defects.
QC is detecting defects.

QA is process related.
QC is product oriented.

QA is for the entire life cycle.
QC is for testing part of SDLC.

QE ---> Quality Engineering

Software Engineer/SE (Write code for developing software)
Quality Engineer/QE (Write code for testing software) (Automation Testers)

Levels of Software Testing
1. Unit Testing
2. Integration Testing
3. System Testing
4. User Acceptance Testing (UAT)

Unit Testing
- A unit is a single component or module of a software.
- Unit testing is conducted on a single program or single module.
- Unit Testing is white box testing technique.
- Unit Testing is conducted by the developers.

Unit Testing techniques:

- Basis path testing (test if every line of code is executed)
- Control structure testing
  - Conditional coverage
  - loops coverage
- mutation testing (testing code with multiple set of data)

Integration Testing

- Integration testing performed between 2 or more modules.
- Integration testing focuses on checking data communication bewteen multiple modules.
- Integrated testing is white box testing technique.

Types of Integration testing:

1. Incremental Integration Testing
2. Non Incremental Integration Testing
3. Sandwich Approach

Incremental Integration Testing

- Incrementally adding the modules and testing the data flow between the modules

There are two approaches:

- Top down approach

- Incrementally adding the modules and testing the data flow between the modules. And ensure the module added is the child of the previous module.

eg - In gmail,
compose mail -> sent items -> deleted items

- Bottom Up approach

Incrementally adding the modules and testing the data flow between the modules. Ensure the module added is the parent of the previous module.

- Sandwich Approach (Hybrid Approach)

Combination of both

Non Incremental Integration Testing

- Integrating all modules at once

Drawbacks:

- We might miss data flow between some of the modules.
- If you find any defect we can't understand the root cause of the defect.

System Testing (will be discussed in detail in later sessions)

- Testing overall functionality of the application with respective client requirement
- It is a black box testing technique.
- This testing is conducted by the testing team.
- After completion of component and integration level testing we start system testing.
- Before conducting system testing we should know the customer requirements.

- System Testing focuses on below aspects:
  - User Interface Testing (GUI)
  - Functional Testing
  - Non-Functional Testing (Installation testing, Compatibility testing, performance testing, security testing)
    (separate dedicated testers are used)
  - Usability Testing
    (documents, user manual, help menu in Apps, user friendliness)

User Acceptance Testing (UAT):

After completion of system testing UAT team conducts acceptance testing in twe level
-Alpha testing (Testing in dev environment)
-Beta Testing (Testing in live environment)

System Testing

Done in tester environment (replica of customer environment)

- qa environment, qc environment, staging environment
- GUI Testing
- Usability Testing
- Functional Testing
- Non-Functional Testing

GUI Testing

- testing the user interface of an application
- tests elements of UI
  - checkboxes, input field, scrollbars, images, colors, sizes, icons, menus, buttons

GUI Testing Checklist

- Testing the size, position, width, height of the elements.
- Testing of the error messages that are getting displayed.
- Testing the different sections of the screen.
- Testing of the font whether it is readable or not.
- Testing the screen in different resolutions with the help of zooming in and zooming out.
- Testing the alignment of the texts and other elements like icons, buttons, etc. are in proper or not.

- Testing the colors of the fonts.
- Testing whether the image has good clarity or not.
- Testing the alignment of the images.
- Testing of the spelling.
- The user must not get frustrated while using the system interface.
- Testing whether the interface is attractive or not.
- Testing of the scrollbars according to the size of the page if any.
- Testing of the disabled fields if any.
- Testing the size of the images.
- Testing of the headings whether it is properly aligned or not.
- Testing the color of the hyperlink.
- Testing UI Elements like button, textbox, text area, check box, radio buttons, drop downs, links etc.

Usability Testing

(provides user manuals / help menu)

- During this testing validates whether the application provided context sensitive help or not to the user.
- Checking how easily the end users are able to understand and operate the application is called usability testing.

Functional Testing

- functionality is nothing but behavior of application.
- Functional Testing talks about how your feature should work.

- Object Properties Testing
- Database Testing
- Error Handling
- Calculations/Manipulations Testing
- Links Existence and Links Execution
- Cookies and sessions

Object Properties Testing

Check the properties of objects present on the Application.

EG: in Textbox properties are enabled and disabled
selection of one radio button at a time
selection of one item in dropdown
selection of multiple item in list boxes
change of focus to next input after insertion of text in a text box

Database Testing (Backend Testing)
(Require some sql commands)
(huge testing step)

As a tester we focus on DML operations

At the beginning, verify the required data coming to the UI.

DML (Data Manipulations Language)
Create, Read, Update, Delete

Work on UI, and verify data in Database

Database testing includes both white box testing and black box testing.
Also called Gray box testing. (Asked in interviews)

Apart from this
- table and column level validation is also done.
  - Column Type, column length, number of columns....
- Relation between the tables (Normalizations)
- Functions
- Procedures
- Triggers
- Indexes
- Views
- etc....

Done by database experts

Error Handling Testing
- Tester verifies the error messages while performing the incorrect actions on the application.
- Error messages should be readable.
- User understandable language./Simple language.

Calculations/Manipulations Testing:
- Tester should verify the calculations.

Links Existence and Links Execution
- where exactly the links are placed --------- Links Existence
- Links are navigating to the proper page or not.

Internal Links
- navigate to same page to different sections

External Links
- navigate to different page
Broken Links
- does not have any target page

Cookies and Sessions
Cookies - Temporary files created by the browser while browsing the pages through the internet.

Sessions are time slots created by the server. Session will be expired after some time (If you are idle for some time)

cookies are created at the browser side.
sessions are created at the server side.

auto logout if application is not used.
used for security mechanisms.

Non Functional Testing
- Done once the application is functionally stable.
- Focus on performance, load it can take and security etc.
Mainly tested as per customer expectation

- Performance Testing --- speed of the application
  - Load Testing
  - Stress Testing
  - Volume Testing
- Security Testing
- Recovery Testing
- Compatibility Testing
- Configuration Testing
- Installation Testing
- Sanitation/Garbage Testing

- Performance Testing --- speed of the application
(done on web based application)
- Load: Increase the load on the application slowly then check the speed of the application.
- Stress: suddenly increase/decrease the load on the application and check the speed of the application
- Volume: Check how much data is able to handle by the application

- Security Testing:
- How secure our application
- mainly focus on Authentication and Authorization

- Authentication ----> User are valid or not
- Authorization/Access Control ---> permissions of the valid user.

Other testing used pen testing, data communication, protocol testing, hacking will be done by security team.

- Recovery Testing:
- checks the system changes from abnormal to normal.

- Compatibility Testing:
- Forward Compatibility
- Backward Compatibility
- Hardware Compatibility (Configuration Testing)

- Installation Testing:
- Check whether the screens are clear to understand or not.
- Screens navigation
- Simple or not.
- Uninstallation

- Sanitation/Garbage Testing:
- Extra features not mentioned in the requirement are also bugs and should be removed.

Regression Testing
- Testing conducts on modified builds to make sure there will not be impact on existing functionality because of changes like adding/deleting/modifying features.

Unit Regression Testing:
- Testing only the changes/modifications done by the developer.

Regional Regression Testing:
- Testing the modified module along with the impacted modules.
- Impact Analysis meeting to identify impacted modules with QA & Dev.

Impact Analysis (Interview Question)
- During impact analysis meetings, we identify impacts because of adding, deleting, modifying functionality or bug fixes.

Full Regression:
- Testing the main feature and remaining part of the application.
- Ex: Dev has done changes in many modules, instead of identifying impacted modules, we perform one round of full regression.

Re-Testing

- Whenever the developer fixed a bug, tester will test the bug fix is called retesting.
- Tester closes the bug if it worked otherwise re-open and send to developer.
- To ensure that the defects which were found and posted in the earlier build were fixed or not in the current build.
- Example:
  - Build 1.0 was released. Test team found some defect (Defect id 1.0.1, 1.0.2)
  - Build 1.1 was released, now testing the defects 1.0.1 and 1.0.2.

Example: Retesting Vs Regression Testing

- An application under test has three modules namely Admin, Purchase and Finance.
  Admin ---> Purchase ---> Finance

- Finance module depends on Purchase.
- If a tester found a bug on the Purchase module and posted. Once the bug is fixed, the tester needs to do Retesting to verify whether the bug related to the purchase is fixed or not and also the tester needs to do regression testing to test the finance module which depends on the purchase module.

Smoke Testing and Sanity Testing

- Smoke and sanity testing comes into the picture after the build release.

Smoke Testing

- Build verification test.
- Smoke Test is done to make sure the build we received from the development team is testable/stable or not.
- Smoke Testing is performed by both Developers and Testers.
- Smoke Testing builds may be either stable or unstable.
- It is done on initial builds.
- It is a part of basic testing.
- Usually it is done every time there is a new release.

Sanity Testing

- Sanity Test is done during the release phase to check for the main functionalities of the application without going deeper.
- Sanity Testing is performed by Testers alone.
- Sanity Testing build is relatively stable.
- It is done on stable builds.
- It is a part of regression testing.
- It is planned when there is not enough time to do in-depth testing.

Exploratory Testing
- We have to explore the application, understand completely and test.
- Understanding the application, identifying all possible scenarios, documenting it then using it for testing.
- We do exploratory testing when the application is ready but there is no requirement.
- Test engineers will do exploratory testing when there is no requirement.

Drawbacks:
- You might misunderstand any features as a bug (or) any bug as a feature since you do not have requirement.
- Time consuming.
- If there is any bug in the application, you will never know about it.

Adhoc Testing
- Testing applications randomly without any test cases or any business requirement document.
- Adhoc testing is an informal testing type with an aim to break the system.
- Tester should have knowledge of the application even though he doesn't have requirements/test cases.
- This testing is usually an unplanned activity.

Monkey/Gorilla Testing
- Testing applications randomly without any test cases or any business requirement document.
- Adhoc testing is an informal testing type with an aim to break the system.
- Testers do not have knowledge of applications.
- Suitable for gaming applications.

Positive Testing
- Testing the application with valid inputs is called Positive Testing.
- checks whether an application behaves as expected with positive inputs.

Negative Testing
- Testing the application with invalid inputs is called Negative Testing.
- checks whether an application behaves as expected with negative inputs.

Requirements:
- For example if a text box is listed as a feature and in FRS and in FRS it is mentioned as Text box accepts 6 - 20 characters and only alphabets.

Positive Test Cases:
- Text box accepts 6 characters.
- Text box accepts upto 20 characters.
- Textbox accepts any value between 6-20 char length.

- Text box accepts all alphabets.

Negative Test Cases:
- Text boxes should not accept less than 6 characters.
- Text boxes should not accept more than 20 characters.
- Text boxes should not accept special characters.
- Text boxes should not accept numbers.

End-To-End Testing
- Testing all over all functionalities of the system including the data integration among all the modules is called end-to-end testing.

Login--->Add Customer--->Delete Customer--->logout
--->Edit Customer

Globalization and Localization Testing

Globalization/Internationalization(I18N) Testing
- Performed to ensure the system or software application can run in any culture or local environment.
- Different aspects of the software application are tested to ensure that it supports every language and different attributes.
- It tests the different currency formats, mobile number formats and address formats are supported by the application.
- For example, Facebook.com supports many languages and it can be accessed by people of different countries. Hence it is a globalized product.

Localization Testing
- Performed to check system or software application for a specific geographical and cultural environment.
- Localized products only support the specific kind of language and are usable only in specific regions.
- It tests whether the specific currency format, mobile format and address format is working properly or not.
- For example, Baidu.com supports only the Chinese language and can be accessed only by people of a few countries. Hence it is a localized product.