

Testing types review 2



SDLC'S which steps are included?

Process name?

Test what?

What action happen? OR
What kind of tests will be done?

Requirement Gathering

Designing

Verification Process

Testing documents

Called : **Static Testing**

- Review (Coding & Testing steps also review documents)
- Walkthrough
- Inspection

Developing /Coding

Testing

Validation Process

Testing the application/code

Called : **Dynamic Testing**

- Unit testing
- Integration Testing

White box testing method

- System Testing
- UAT

Black box testing method

Levels of software testing

- Level one: Unit testing
- Level two: Integration testing
- Level three: System testing
- Level four: User Acceptance Testing(UAT)

Unit Testing example:

```
public int CitiesInState(String state){  
    //TODO  
    int count =0;  
    for(int i =0; i<=cityList.length-1; i++) {  
        if(cityList[i].state.equalsIgnoreCase(state)) {  
            count++;  
        }  
    }  
    return count;  
}
```

This is just one unit of the code.
Test only this single part is called **Unit Testing**

```
public int population(String city){  
    //TODO  
    for(City k : cityList) {  
        if(k.name.equalsIgnoreCase(city)) {  
            return k.population;  
        }  
    }  
    return -1;  
}
```

← One Unit

```
public String state(String city){  
    //TODO  
    for(City k : cityList) {  
        if(k.name.equalsIgnoreCase(city)) {  
            return k.state;  
        }  
    }  
    return null;  
}
```

← One Unit

In this example, total 3 unit testing can be done by Developers

Integration Testing example:

```
public int CitiesInState(String state){  
    //TODO  
    int count =0;  
    for(int i =0; i<=cityList.length-1; i++) {  
        if(cityList[i].state.equalsIgnoreCase(state)) {  
            count++;  
        }  
    }  
    return count;  
}
```

When a user enter a state name, the code will print all cities in that State.

Input : State
Output : Cities

```
public int population(String city){  
    //TODO  
    for(City k : cityList) {  
        if(k.name.equalsIgnoreCase(city)) {  
            return k.population;  
        }  
    }  
    return -1;  
}
```

If we take a city's name from above code, and enter to this code, it will print the population.

Input : a city's name
Output : Population

Integration testing :

Combination of above 2 unit tests , take the 1st code's output as the second code's input, and check if they work together or not.

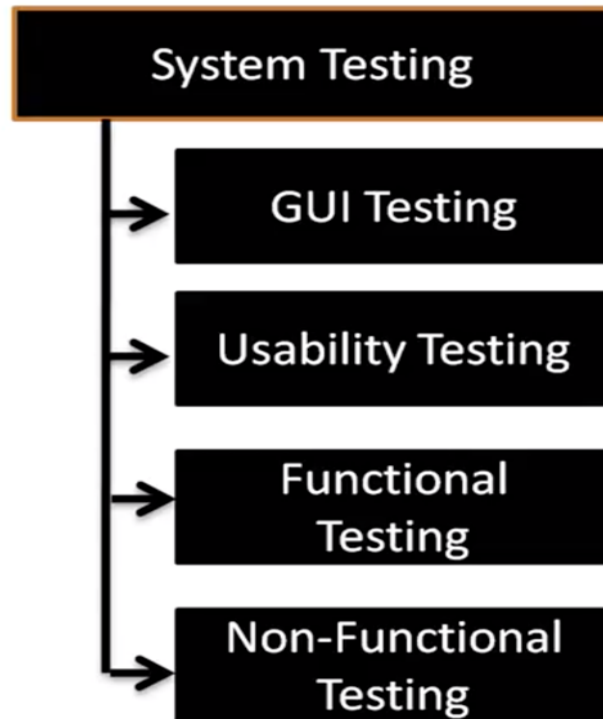
(Can combine more then 2 unit tests)

Unit testing & Integration testing

- performed by Developer → to develop the application
- An automation engineer may perform Integration testing. (It depends the tester's ability)

3rd level of software testing : System testing

System Testing Types



GUI Testing



Graphical User Interface testing



Mainly test the application's:

Spelling error

Color

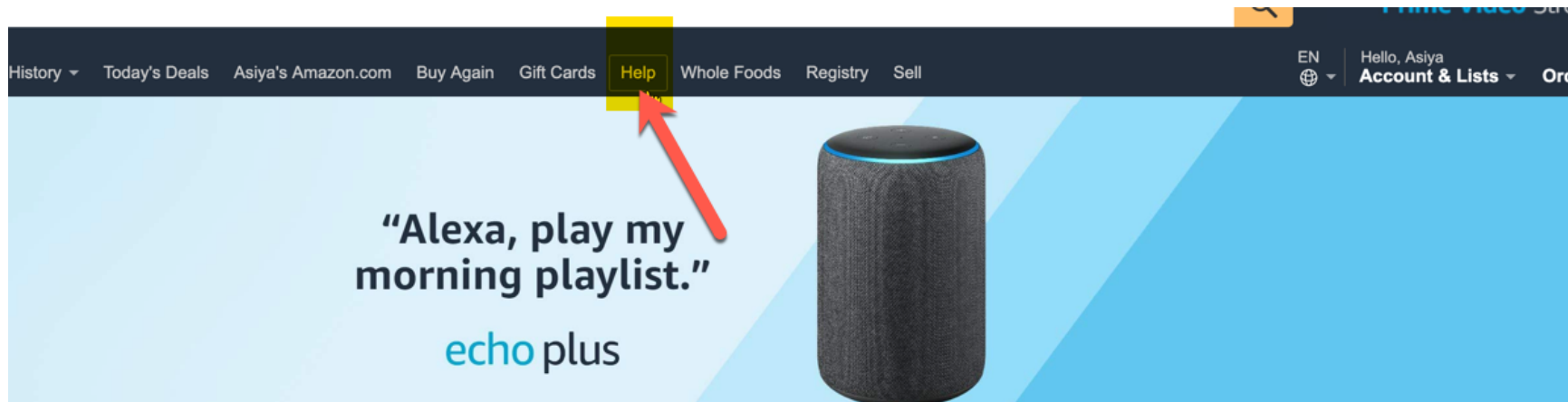
error messages Format

Size of an image

Etc.

Usability Testing

- evaluates the application on how easy it is for use it.
- Check if the application is provided “**Help**” or not.
- It can be a module of the application, or separate documentation.



Functional testing

- Functionality testing is performed to verify that a software application performs and functions correctly.
- What is a functionality of an application/project?
 - Login
 - Logout
 - Order food
 - Search food
 - Do payment
 - write comments

Functional testing

- Types of functional tests:
 - Smoke test
 - Regression test
 - User Acceptance Testing(UAT)
 - Etc.

(No worries about the rest, these are the most important test you will do as an automation engineer)

Smoke testing

- Example:
 - We have **150 user stories** in product backlog, and all developed and tested.
 - Users/customers are using the Cyberburger website.
 - Customers can *register*.
 - Customers can *login*.
 - Customers can *see menu*.
 - Customers can *order food*.
 - Customers can *logout*.
 - Customers can *do payment*.
 - Customers can *contact someone for help*.
 - Customers can *receive conformation emails*.
 - Customers can *write comments*.

- In order to make sure there was no any issue happed during the night time
- every morning before scrum team start to work, before any customer go to the website and use it
- around 6 AM(It depends)
- Only **the major functionalities** of the application will be tested automatically with the help of some tools.

Let's say 7 user stories out of 150 tests every day which include the main functionalities:

- Login
- order food
- see menu
- *do payment.*

Regression testing

→ Whenever a new functionality added to the application, we have to make sure it shouldn't affect the existing functionalities/features.

- Let's say we are in **sprint 20**:
- So far we finished 150 user stories:
- In the new sprint, we will add one more functionality, and we will **add 5** more User stories
 - As a user, I should be able to collect my favorite foods and drinks in “**Favorites**” **module**.

Regression testing

We add “Favorites” functionality, login-logout, food ordering and all existing functionalities should work as expected.

So we test all **150 + 5** user stories at the same time.

(Some companies only choose half of them as regression because it takes a long time to test or they do not have enough automation tester, or no tools to help running regression faster)

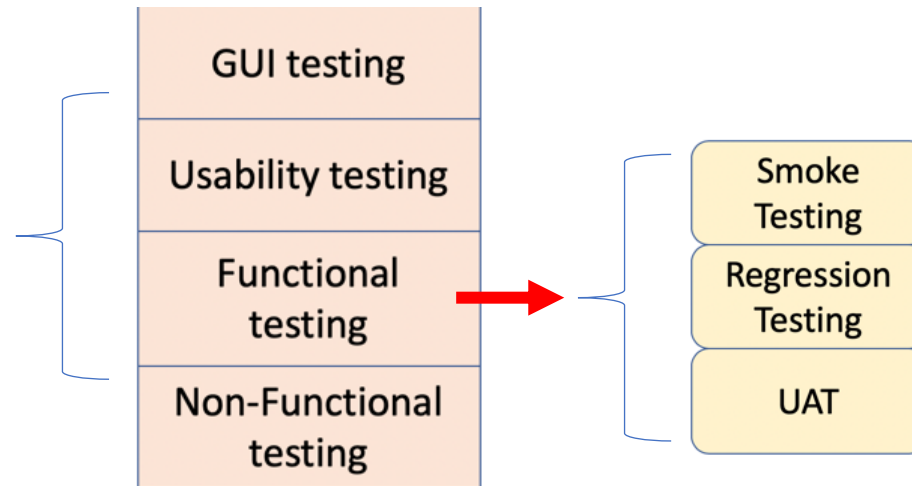
Level's of software testing

Unit Testing

Integration Testing

System Testing

UAT



Team lead



developer developer developer

Dev-team

Team lead



Tester Tester

Testing - team

Non-Functional testing

- Types of Non-functional testing :
 - Performance testing. → mainly test the speed of the application.
 - Load Testing
 - Stress Testing
 - Volume Testing
 - Security Testing
 - Installation Testing
 - Etc.

Non-Functional testing → Performance testing

Load Testing

- 10 costumer use the Cyberburger
- 20 costumer use at the same time
- every time increase the number of user a little bit

Observe/ test if the application is become slower or not.

Stress Testing

Test the respond of the application under the stress.

In this testing, **immediately** increase or decrease load, or number of user, from 10 to 90, and give stress to the application.

Volume Testing

How much of data that the application is able to handle.

Non-Functional testing

- An IT Company has a team called “***Performance team***”
- *Performance team responsible for non-functional testing.*