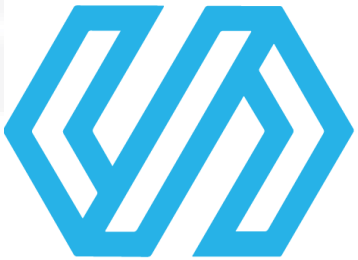
Types of testing



SKILLO

Agenda

1. According to the knowledge of the system
2. According to the object of testing
3. According to the time of testing
4. By degree of automation
5. On the basis of a positive scenario
6. Degree of readiness for testing
7. According to the degree of isolation of components

SKILLO

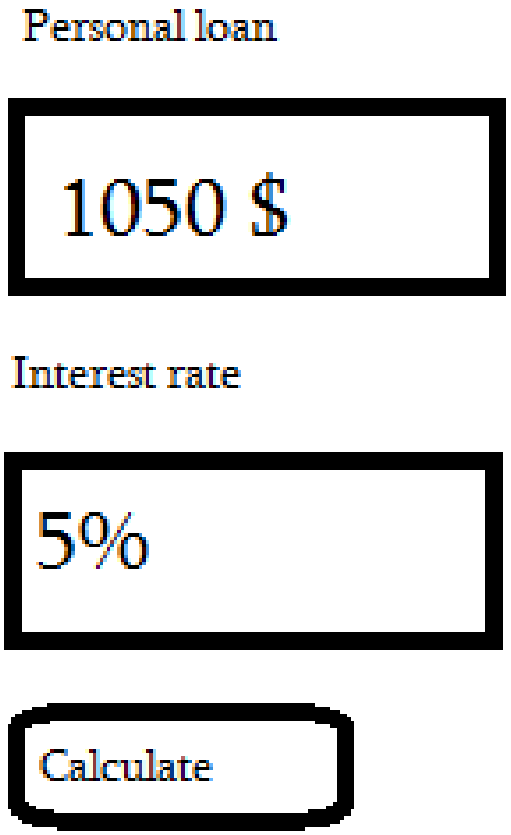
According to the knowledge of the system

Black box testing

- ✓ testing, functional or non-functional, without any knowledge of the internal
- ✓ structure of a component or system
- ✓ Based on the functionality and the requirements of the system
- ✓ What the system is doing
- ✓ Most of the applications are covered mainly by black box testing
- ✓ Tester need not know programming languages

SKILLO

Black box testing example



SKILLO

According to the knowledge of the system

White box testing

- ✓ testing based on an analysis of the internal structure of a component or system
- ✓ How is the system doing it
- ✓ Known as Glass box or Clear box testing
- ✓ Programming know-how and the implementation knowledge is essential
- ✓ Mainly applied to Unit testing
- ✓ We not need the GUI to design the tests
- ✓ QA engineers with programming knowledge are needed

SKILLO

White box testing example

```
var loanAmount = args[0];
var interestRate = 0;

if(loanAmount == 0){
    showRate(interestRate);
} else if(loanAmount <= 1000){
    interestRate = 6;
    showRate(interestRate);
} else if(loanAmount > 1000 and loanAmount < 2000){
    interestRate = 5;
    showRate(interestRate);
} else if(loanAmount > 2000 and loanAmount < 10000){
    interestRate = 3;
    showRate(interestRate);
} else {
    interestRate = 2;
    showRate(interestRate);
}
```

SKILLO

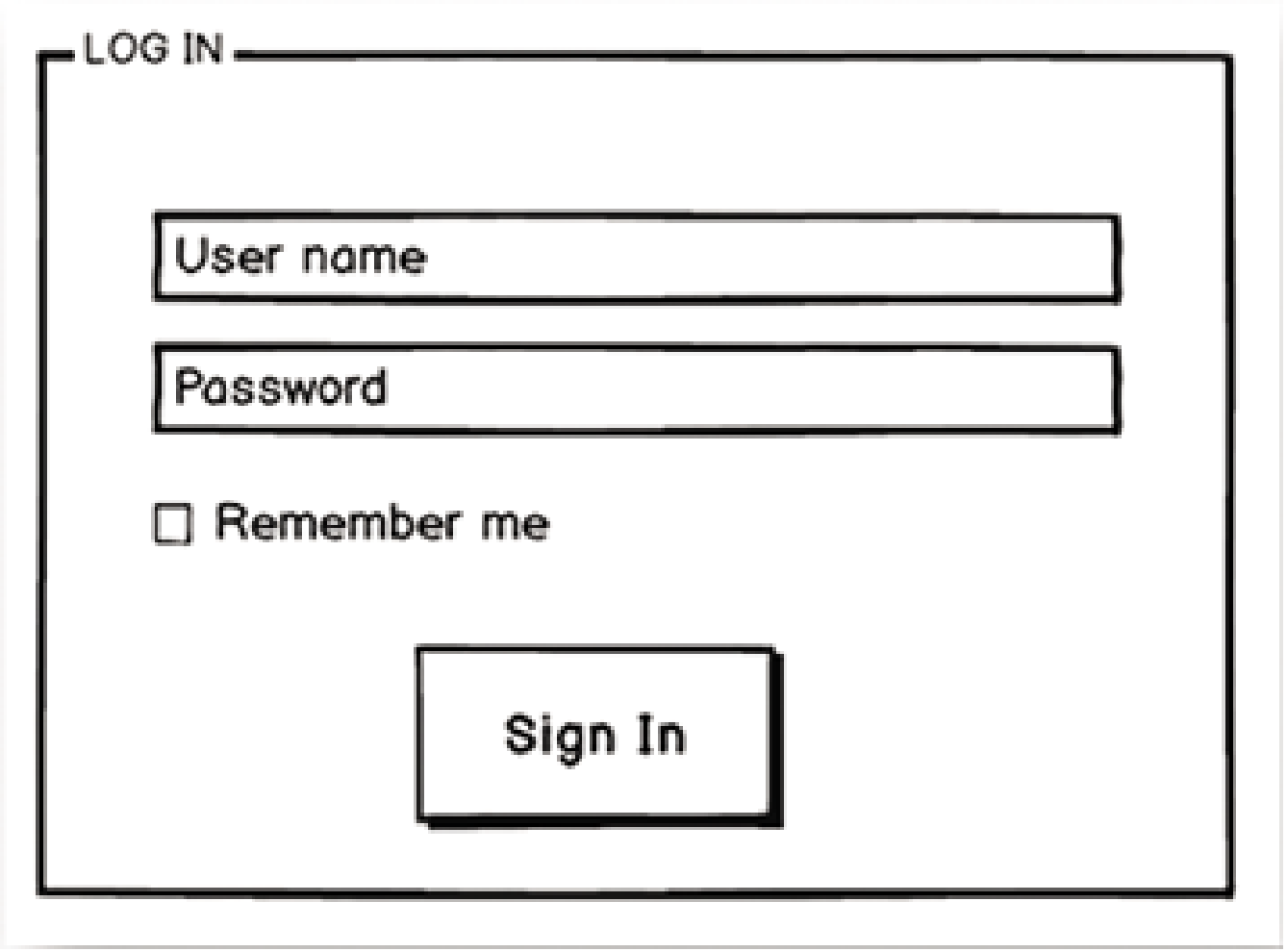
According to the knowledge of the system

Gray box testing

- ✓ Combination of black box and white box testing
- ✓ Very effective
- ✓ QA creates the tests based on the functionality but in addition execute checks in the code

SKILLO

Gray box testing example



SKILLO

According to the object of testing

Functional testing

- ✓ based on an analysis of the specification of a component or system functionality
- ✓ most popular testing

Example:

1. Go to login page www.nonexistingpage.com
2. Click sign in
3. Fill username and password
4. Click login button

Expected result:

The user is redirected to the Homepage and sees his/her user profile picture and name

SKILLO

According to the object of testing

UI testing

- ✓ requires verification of compliance requirements of the application to the graphical interface. Whether SUT is professional looking, use the same style etc.

According to the object of testing

Localization testing

- ✓ Verify UI in case of different translations
- ✓ Handling of different input text encoding

Example:

If our site is translated on English, Bulgarian and Chinese Language. We should check if we can login/create user etc. using Latin, Cyrillic or Chinese using our site with the relevant language

According to the object of testing

Usability testing

- ✓ Performed to the perspective of the client
- ✓ Checks if GUI is user-friendly
- ✓ Is there learning curve using the SUT
- ✓ Black box testing

Example:

Testing travel site we should test what is the user experience for the main scenarios. Book flight, book hotel, rent car etc.

According to the object of testing

Installability testing

- ✓ Checks if the installation document is suitable for installing the application into environment properly or not

We should consider the tests below:

- ✓ Install the software on clean machine
- ✓ Upgrade of already existing version
- ✓ Uninstall the software
- ✓ Not enough disk space during the installation
- ✓ OS not supported

According to the object of testing

Security testing

- ✓ We test security of the SUT. This means we check for SQL injections, code injections, conformity to the security standard ISO 27001, data protection etc.

According to the object of testing

Recovery testing

- ✓ check how fast and better the application can recover after it has gone through any type of crash or hardware failure etc.
- ✓ Data loss and time to restore from system failure

According to the object of testing

Compatibility testing

- ✓ verify the correct operation of the product in a particular environment (different hardware, OS, network, databases, browsers)

Example:

- ✓ Special requirements for a user's video cards (hardware)
- ✓ Mobile apps
- ✓ When a Web site uses some cutting-edge scripting technology that might be treated differently by different Web browsers (css , java script)

SKILLO

According to the object of testing

Performance testing

- ✓ the process of testing to determine the performance of a software product
- ✓ check the response time of our Web site or its components
- ✓ Find and remove/workaround **bottlenecks**

Performance testing should have requirements:

- ✓ Response time with 1000 users
- ✓ Simultaneous users per minute
- ✓ Maximum number of users
- ✓ Maximum loading time of a page etc.

SKILLO

According to the object of testing

Load testing

- ✓ Checks how the system behaves under different loads
- ✓ Analysing response times under different loads

SKILLO

According to the object of testing

Stress testing

- ✓ Tests the system beyond the performance boundaries described in the specification
- ✓ Idea is to check how the system behaves under unexpected/unusual performance conditions

According to the object of testing

Static testing

- ✓ Checks component or system without execution of software code, such as reviewing or static analysis

SKILLO

According to the object of testing

Dynamic testing

- ✓ Testing is conducted during the execution of SUT

SKILLO

According to the time of testing

Alpha testing

- ✓ Testing by potential internal users
- ✓ It is internal acceptance testing
- ✓ Real customers or independent test team

Beta testing

- ✓ Testing by potential or existing clients
- ✓ Do not contact developers during testing
- ✓ It is external acceptance testing
- ✓ Gives a real market feedback

SKILLO

According to the time of testing

Regression testing

- ✓ Testing already tested program, after modifications
- ✓ make sure that the process of modification has not made mistakes or not activated in areas not subject to change
- ✓ Held after the changes in the code of the software or its environment.

Acceptance testing

- ✓ Formal testing by the clients, PM, Business owners to determine whether or not to accept the system
- ✓ Usually straight case (business) scenarios are executed

SKILLO

According to the time of testing

Smoke testing

- ✓ Covers the basic functionality of the SUT
- ✓ Executed before the detailed testing to check if it makes sense to continue

By degree of automation

Manual Testing

- ✓ Test cases and test data are created and executed manually

Automation Testing

- ✓ Test cases are executed by automation testing tools
- ✓ Automation test cases are created manually
- ✓ Perfect for regression testing

Semi-automation Testing

- ✓ Automation tools or scripts are used to help manual testing
- ✓ Test data can be generated
- ✓ Common steps can be automated

On the basis of a positive scenario

Positive Testing

- ✓ Verify the system functions/behave as expected
- ✓ Covers the use case scenarios

Negative Testing

- ✓ How system behaves in case of errors or irregular use
- ✓ A lot of negative combinations are available
- ✓ Negative testing founds more bugs

Degree of readiness for testing

Formal Testing

- ✓ Testing based on the designed test cases
- ✓ Executed once the software is ready for testing

Exploratory Testing

- ✓ Informal type of testing
- ✓ Design the tests during the execution
- ✓ Help to generate new and better tests

Ad Hoc Testing

- ✓ Informal type of testing
- ✓ Based on the experience of the QA
- ✓ Test cases are not designed and recorded

According to the degree of isolation of components

Component (Unit) testing

- ✓ Testing components of the software in isolation

Integration Testing

- ✓ Testing the interaction between integrated components or systems

System Testing (End to end testing)

- ✓ Testing of the SUT as a whole
- ✓ Exercises the business scenarios
- ✓ Usually longer and complex scenarios

Component testing example

```
public static int sumTriangleSides(decimal a, decimal b, decimal c){
    .......
    .......
    .......
}

[TestFixture]
public class TriangleTests
{
    [Test]
    public void SumWithThreePositiveNumbers()
    {
        decimal sum = sumTriangleSides(3.0, 4.0, 5.0);
        Assert.AreEqual(12, sum);
    }

    [Test]
    public void SumWithNegativeNumber()
    {
        decimal sum = sumTriangleSides(3.0, -4.0, 5.0);
        Assert.AreEqual(null, sum);
    }
}
```

SKILLO

Integration testing example

Component1 – Login
Component2 - Basket
Component3 - Payment

Component1 interaction with Component2

Component1 interaction with Component3

Component2 interaction with Component3

System testing example

Component1 – Login
Component2 - Basket
Component3 – Payment

Whole scenario
Starts with Component1 and goes to the end
Conduct the whole process

Q & A

SKILLO

THANK YOU

SKILLO