**Занятие 6**

Типове тестване:

1. Според познаване на системата.

2. Според обекта на тестване

- functionality

- UI

- localization

- usability

- instalability

- compatibility

- security

- recovery

- performance

- load

- stress

- static

- dynamic

3. Според времето за тестване

- alpha

- beta

- regression

- acceptance

- smoke

- sanity

4. By degree of automation

- manual

- automatick

- semi-automatick

5. На база на позитивен сценарии

- positive

- negative

6. Степен на готовност за тестване

- formal

- exploratory

- ad hoc

7. Според степента на изолация на компонентите

- component (unit)

- integrasion

- system (end to end)

––––––––––––––––––––––––––––––––––––––––––

**1. Според познаване на системата**

Black box тестване

- Тестване, функционално или нефункционално, бес каквото и да е знание на вътрешноста на система

( пример: функционално – как калкулатора пресмята /

нефункционално – дали формата и цвета на калкулатора са според спецификациата )

- Структура на компонент или система

- Базирано на функционалноста или изискванията на системата

- Какво прави системата

- Почти фсички приложения се покриват от блек бокс тестване

- Тестъра трябва да знае програмни езици

White box

- Тестване базирано на анализ на вътрешната структура на компонента или системата

- How the system is doing it

- Познато още като Glass box or Clear box тестване

- Programin know-how and the implementation knowlege is essencial

- Mainly applied to Unit testing ( Unit – the smallest component of a system)

- QA engineers with programming knowlege are needed

Gray box testing

- Very effective

- Combines black and white box testing

- QA creates the test based on specification but in addition execute checks in the code

**2. According to the object of testing**

Functional testing

- Based on an analysis of the specification of a component or system functionality

- Most popular testing

Exemple:

Go to www.somesite.com

Click the sing in

Fill username / password

Click login button

Expectd result:

The user is redirected to homepage and sees his/hers profile picture and name

UI (user interface) testing

- If the UI coresponds to the cecification / mockups, have consistent style etc.

Localization testing ( languages)

- Verify UI in case of different translations

- Handling of different input text encoding

Example:

If our site is translated to bulgarian, english, chinese language. We should check if user can login / create account etc using the different languages

Usability testing

- Performed to the perspective of the client

- Check if GUI (graphic user interface) is user-frendly

- Is there learning curve to using the SUT

- Black box testing

Example:

Testing travel site we should test what is the user experiance for the main scenarios

Book flight, book hotel, rent car etc.

Instalability testing ( if software can instal and uninstal from the targeted platform / OS)

- Check if the installation documantation is suitabel for installing the application into enviroment properly or not

We should consider testing

- Install the software on clean machine

- Upgrade of alredy existing version

- Uninstall the software

- Noty enough disk space during the instalation / upgrade

- OS not supported

Security testing

- We test security of the SUT. This means we check for SQL injections, code injections, conformity to the security standard ISO 27000, data protection etc

Recovery testing

- Check how fast and better the application can recover after it has gone trough any type of crach or harduer failure tec.

- Data loss and time to restore after system failure

Compatibility test

- Verify the correct operation of the product in a particular enviroment

(different hardware, os network, database, browser)

Performance test

- the process of testing to determine the perrformance of a software product

- check the responce time of our Web site or its components

- find and remove / workaround bottlenecks

Performance testing should have requirments:

- responce time with 1000 users

- simultaneous users perrformance minute

- maximum number of users

- maximum load time of a page etc.

Load testing

- Check how the system behaves under different loads

- Analysiing response times under different loads

Stress testing

- Test the system beyond the performance bounderies describet in the specification

- idea is to check how the sistem behaves under unexpected / unusual performance conditions

Static testing

- Check components or systems without execution of the software code, such as reviewing or static analysis

Dynamic testing

- runing the code and see where ib breacks (official: testing is conducted during the execuition of SUT)

**2. Acording to the time of testing**

Alpha testing

- testing by potencial internal users

- it is internal acceptance test

- real customers or independant testing team

Beta testing

- gives real market feedback

- testing by potencial or real clients

- do not contact developers

- it is external acceptance tes

Regression testing

- after adding new features the whole system is retested to make sure nothing breaks by the new addition and everything work

official:

- testing alredy tested program, after modifications

- make sure that the procces of modification has nothing made mistakes or nothing activated in the areas nothing subject to change

- held after the changes in the code of the software or its enviroment

Sanity check (test)

- the point of sanity check is to rule out certain classes of obviously false results, not to catch every possible error.

- the advantage of sanity test is speed

Acceptance test

- formal testing by the clients, business owners to determine wheter or not to acceptance the system

- usualy straigh case (business) scenarios areas executed

Smoke testing

- covers the basic functionality of the SUT

- executed before the ditailed tested to see if it makes sense to continue

**4. By the level of automation**

Manual testing

- tested cases and tested dat

Automatic testing

- test cases are executed by automation tools

- automation test cases are created manually

- perfect for regression testing

Semi-automatic testing

- automation tools or scripts are used to help manual testing

- test data can be generated

- common steps can be automated

**5. On basis of positive scenario**

Positive testing

- verify the system functions/behaviour as expected

- covers the use of case scenarious

Negative testing

- how system behaves in case of error or irregular use

- alot of negative combinations are avalible

- negative testing founds more bugs

**6. Degree of rediness for testing**

Formal testing

- testing based on designed test cases

- Executed once the software is redy for testng

Exploratory testing

- informal type of testing

- design the tests during the execution

- help to genarate new and better tests

ad hoc test

- прави се преди acceptacne теста, когато продукта е на практика готов

- откритите грешки и бъгове се документират, но няма теста кейсове по който да се води

official:

- informal type of testing

- based on experiance of the QA

- test cases are not created and recordet

**7. According to the degree of isolation of componens**

Component (Unit) testing

- testing the components of the system in isolation

integration testing

- testing the interaction of the integrated components in the systems

system testing (end to end testing)

- testing the SUT as whole

- exerceses the business scenarious

- usualy longer and complex scenarios