openIMIS

Testing scenarios of openIMIS

Jean Brice Tetka

20.05.2021

Why testing?

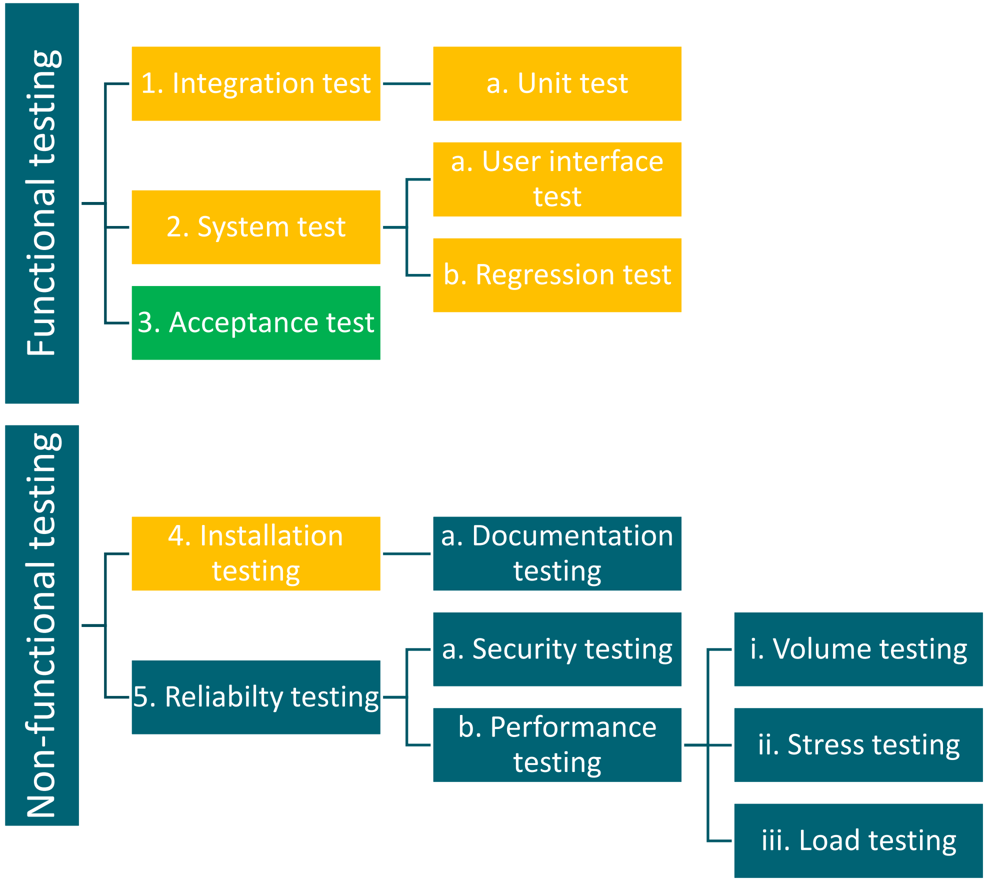
- Testing is an essential step for detecting and preventing bug
- The testing phase will guarantee that the software satisfies the user requirements, with a good quality and reliable product.

Functional testing	**Non-functional testing**
Verifies each function/feature of the software	Verifies non-functional aspects like performance, usability, reliability, etc
Can be done manually	Hard to perform manually
based on customer requirements	Based on customer’s expectations
To validate software actions	To validate the performance of the software
Describes what the product does	How the product works

What if the update of openIMIS stops?

- The maintenance and the update of openIMIS will be significantly affected.
- To mitigate that risk, the openIMIS initiative is facilitating the transfer of knowledge and expertise.
- A proposed approach is to transfer the knowledge on the testing of openIMIS via testing cases and scenarios

Testing scenarios



- Functional testing
  - 1. Integration test
    - a. Unit test
  - 2. System test
    - a. User interface test
    - b. Regression test
  - 3. Acceptance test
- Non-functional testing
  - 4. Installation testing
    - a. Documentation testing
  - 5. Reliabilty testing
    - a. Security testing
    - b. Performance testing
      - i. Volume testing
      - ii. Stress testing
      - iii. Load testing

Scenario 1: A new implementation

Steps	Type of test	Target	Test type	Actors
Before the implementation	1. a. Unit test	Test each unit/module	Manual	Developers
Before the implementation	1. Integration test	Test if the combined modules work together	Manual	Developers
Before the implementation	2.a User interface test	Test if the user interface works as required	Manual	Developers/Users
Before the implementation	2. System test	Test the complete software	Manual	Developers/Users
Before the implementation	3. Acceptance test	Test if the software meets the user requirements	Manual	Users

Scenario 1: A new implementation

Steps	Type of test	Target	Test type	Actors
During the implementation	4. Installation testing	Test if the software is correctly installed	Manual	Developers
During the implementation	4.a. Documentation testing	Test if documentation about how to use the system matches with what the system does	Manual	Developers/Users
During the implementation	5.a. Security testing	Test that security breaches can be prevented	Manual/Automated	Security expert
During the implementation	5.b. Performance testing	Test the speed, stability, and scalability and resource usage under particular workload.	Manual/Automated	Testing expert

Scenario 2: A new release

Steps	Type of test	Target	Test type	Actors
Before the release (if more than one module is updated)	1. a. Unit test	Test each unit/module	Manual	Developers
Before the release (if more than one module is updated)	1. Integration test	Test if the combined modules work together	Manual	Developers
Before the release	2.b. Regression test	Test if the software works properly after a change in a module	Manual	Developers
Before the release	2. System test	Test the complete software	Manual	Developers/Users

Scenario 2: A new release

Steps	Type of test	Target	Test type	Actors
During the release	4. Installation testing	Test if the software is correctly installed	Manual	Developers
During the release	4.a. Documentation testing	Test if documentation about how to use the system matches with what the system does	Manual	Developers/Users

Scenario 3: A bug is found

- **Critical bug**: When critical functionalities or data are affected, the software is at great risk. That can be for example a failure to access the data of beneficiaries or a security breach into the system

- **Major bug**: When critical functionalities or data are affected but not stopping of the use of the platform. For example, if a user is unable to enrol a new beneficiary but can manage the existing list of beneficiaries.

- **Minor bug**: Is bug that affects minor functionalities or non-critical data. Like a field not displaying the entire name of the beneficiary.

- **Trivial bug**: Is bug not affecting any functionality or data, but just causing some inconvenience. Like a wrong translation of a field.

Scenario 3: A bug is found

Bug severity	Type of test	Target	Test type	Actors
Critical	1. a. Unit test	Test each unit/module	Manual	Developers
Critical	1. Integration test	Test if the combined modules work together	Manual	Developers
Critical	2.a User interface test	Test if the user interface works as required	Manual	Developers/Users
Major	2.b. Regression test	Test if the software works properly after a change in a module	Manual	Developers
Critical/Major	2. System test	Test the complete software	Manual	Developers/Users
Critical/Major	4. Installation testing	Test if the software is correctly installed	Manual	Developers
Critical	5.a. Security testing	Test that security breaches can be prevented	Manual/Automated	Security expert
Critical	5.b. Performance testing	Test the speed, stability, and scalability and resource usage under particular workload.	Manual/Automated	Testing expert

What next?

- Refine the scenarios
- Record and document the integration and the system testing
- Define test cases for installation testing
- Work with security testing expert to define testing plan and do the test
- Work with performace testing expert to test the performance of oI

Thank you !