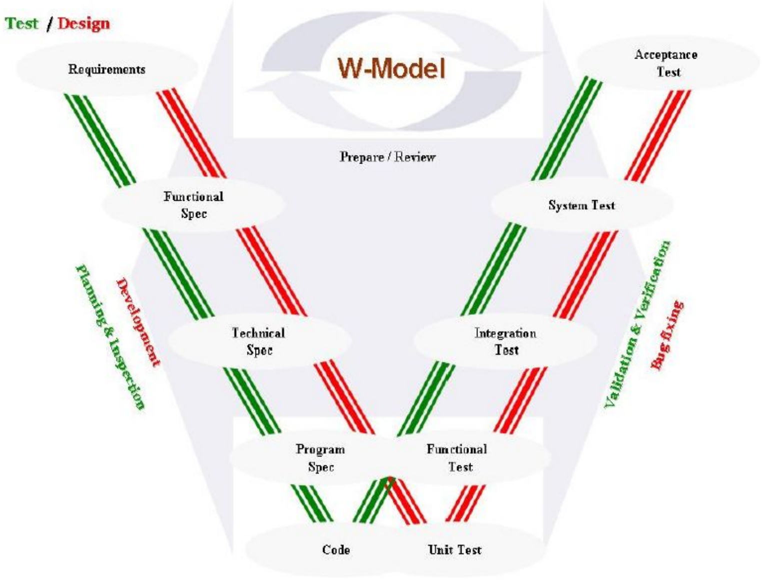
HINTS on Test&QA in SCRUM

Fall 2021

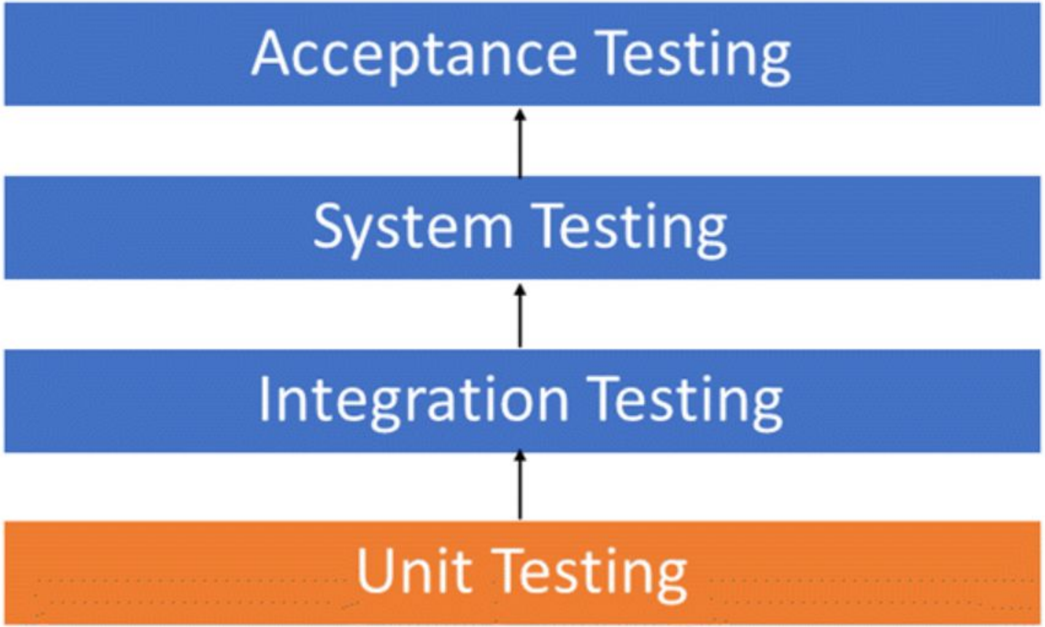
Group A & B & F
Eloi Puertas - epuertas@ub.edu
Eduardo Urruticoechea - e.urruticoechea@ub.edu

TEST CONCEPTS: The W model

The W-Model defines two parallel paths, one for Development (Design) and one for Test, for eac activity in Development; there is a related activity in Test.



TEST CONCEPTS: Types of tests



Acceptance Testing

System Testing

Integration Testing

Unit Testing

Unit Testing Levels

TEST CONCEPTS: Test Driven Development

- TDD are the Unit Tests in the W-model.
- These tests must be performed in the code to avoid all the possible and common mistakes.
- Development team must guarantee that the build can be created, and will work as expected.

TEST CONCEPTS: Unit Testing Tools

1. Junit: Junit is a free to use testing tool used for Java programming language. It provides assertions to identify test method. This tool test data first and then inserted in the piece of code.
2. NUnit: NUnit is widely used unit-testing framework use for all .net languages. It is an open source tool which allows writing scripts manually. It supports data-driven tests which can run in parallel.
3. JMockit: JMockit is open source Unit testing tool. It is a code coverage tool with line and path metrics. It allows mocking API with recording and verification syntax. This tool offers Line coverage, Path Coverage, and Data Coverage.
4. EMMA: EMMA is an open-source toolkit for analyzing and reporting code written in Java language. Emma support coverage types like method, line, basic block. It is Java-based so it is without external library dependencies and can access the source code.
5. PHPUnit: PHPUnit is a unit testing tool for PHP programmer. It takes small portions of code which is called units and test each of them separately. The tool also allows developers to use pre-define assertion methods to assert that a system behave in a certain manner.

Unit Testing in Python

https://realpython.com/python-testing/

Unit Testing Best Practices

- Unit Test cases should be independent. In case of any enhancements or change in requirements, unit test cases should not be affected.
- Test only one code at a time.
- Follow clear and consistent naming conventions for your unit tests
- In case of a change in code in any module, ensure there is a corresponding unit Test Case or the module, and the module passes the tests before changing the implementation
- Bugs identified during unit testing must be fixed before proceeding to the next phase in SDLC
- Adopt a "test as your code" approach. The more code you write without testing, the more paths you have to check for errors.

Testing at Atlassian

- https://youtu.be/yRP29wFqu20