Software Engineering

Class 7

Fall 2023

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

Class SCHEDULE

Class SCHEDULE

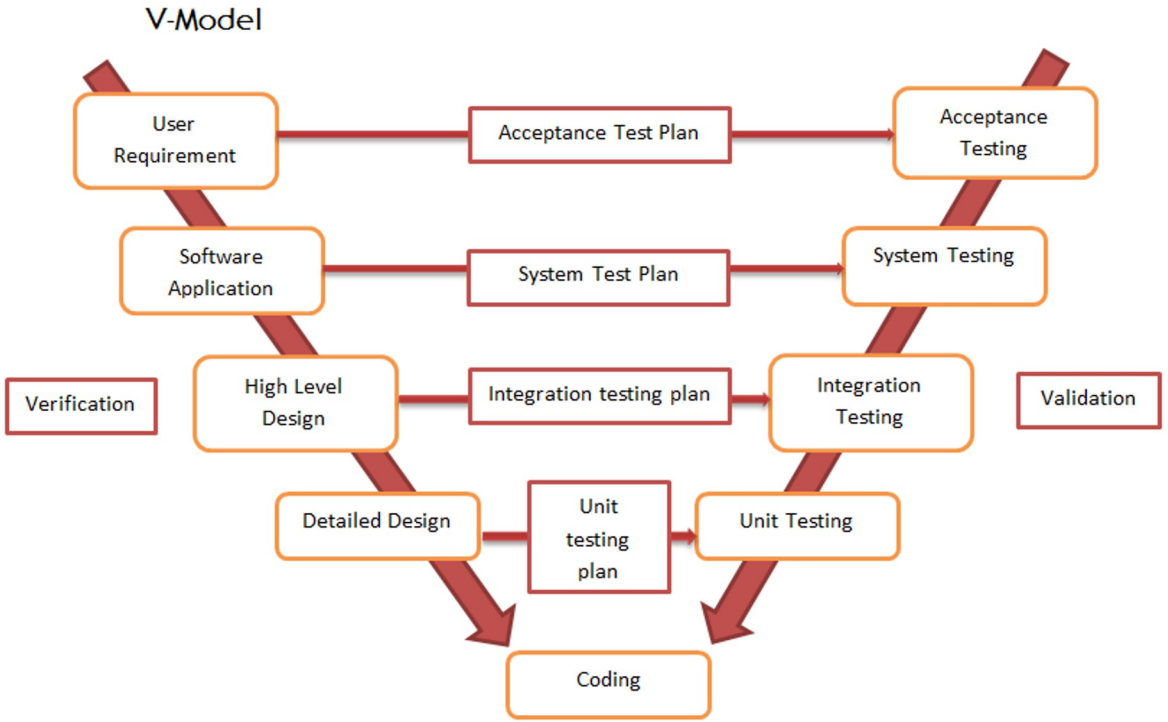
- **[Set 20 & 22]** **Class 1: Project KickOff.**
- **[Set 27 & 29]** **Class 2: Backlog check.**
- **[Oct 4 & 6]** **Class 3: Deliver Backlog. Sprint 0 Planning.**
- *[Oct 11 & 13]* *Bank holidays*
- **[Oct 18 & 20]** **Class 4: Deliver Demo S0. Retrospective Sprint 0. Sprint 1 Planning.**
- **[Oct 25 & 27]** **Class 5: Sprint 1 check [Q1]**
- *[Nov 1 & 3]* *Examen, no class.*
- **[Nov 8 & 10]** **Class 6: Deliver Demo S1. Retrospective Sprint 1**
- **[Nov 15& 17]** **Class 7: Sprint 2 check.**
- **[Nov 22 & 24]** **Class 8: : Deliver Demo S2. Retrospective Sprint 2.**
- **[Nov 20 & Dec 01]** **Class 9 Sprint 3 check [Q2]**
- *[Dec 06 & 08]* *Bank holidays*
- **[Dec 14 & 16]** **Class 10: Deliver Final PRODUCT (S3).**

Sprint 2 check:

1. Focus on QA: Testing and quality assurance beyond unitary testing
2. Focus on RETROSPECTIVE: Self-organizing teams is the fundamental of agile frameworks !

TEST CONCEPTS: The V (or W) model

The V-Model defines two parallel paths, one for Development and one for Test. For each activity in Development, there is a related activity in Test.



TEST CONCEPTS: Unit Tests & Test Driven Development

- TDD are the Unit Tests in the V-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: Integration tests

- Integration tests determine if independently developed units of software work correctly when they are connected to each other.
- The objective is to check if separately developed modules work together properly.
- Integration tests are defined “against” the design and architecture of the system, and should focus in the most critical and overlapped areas of the software design.
- BLACK BOX or WHITE BOX TESTING ?

TEST CONCEPTS: Functional Testing

- The test cases are defined on the specifications of the software component under test.
- Functions are tested by feeding them input and examining the output, and internal program structure is not considered.

Functional testing describes *WHAT* the system does. NOT HOW !

TEST CONCEPTS: TYPES of Functional Testing

- SMOKE TESTING: Preliminary testing to reveal simple failures. Smoke tests are, usually, a subset of functional tests that cover the most important functionality of a component or system, used to aid assessment of whether main functions of the software appear to work correctly.
- REGRESSION TESTING: Re-running functional and non-functional tests (almost ALL) to ensure that previously developed and tested software still performs after a change.
- USABILITY TESTING: A technique used to evaluate a product by testing it on users. This can be seen as an irreplaceable usability practice, since it gives direct input on how real users use the system

TEST CONCEPTS: Other types of tests

- BEHAVIOUR DRIVEN DEVELOPMENT: Evolution of TDD into business and use cases. (See https://cucumber.io/)
- SECURITY TESTING: Process intended to reveal flaws in the security mechanisms of an information system that protect data and maintain functionality as intended. (I.e: elements of confidentiality, integrity, authentication, availability, authorization and non-repudiation).
- EXPLORATORY TESTING: Is an approach to software testing that is concisely described as simultaneous learning, test design and test execution. Is random or unstructured in nature and can reveal bugs that would go undiscovered during structured phase of testing.

Testing and beyond !

- International Software Testing Qualifications Board (ISTQB Certifications)
- https://www.istqb.org/

Tips on QA on every sprint planning

WE ARE NOT GOING TO TEST EVERYTHING….BUT SOME TESTING IS NEEDED, SO YOU MUST PLAN FOR TESTING BECAUSE IT REQUIRES TIME…..THEREFORE:

1. FOCUS on the most critical and complex areas to develop complete tests
2. Select the most appropriate techniques for each area to test.
3. Select the most easy test frameworks to develop the different tests
4. AUTOMATE, automate, automate !

COURSE EVALUATION

1. Sprint 0: Kick-off. Initial team building. Technical decisions. — 10 %
2. Sprint 1: Focus on process and technical stability. — 20%
3. Sprint 2: Focus on testing, QA. Mature DevOps. AND Retrospective self-assessment. — 30%
4. Sprint 3: Focus on final product demonstration. AND Retrospective self-assessment. — 40%

Retrospective improvement

WHAT WE DID WELL ;-) : Have each team member write down what the team did well, one idea per note. Post the notes, and group similar or duplicate ideas together. Discuss each one briefly as a team.

- What have you done really well in the last iteration?
- What is something that makes you really happy?
- What nice thing did you do for someone else last iteration?

WHAT WE CAN IMPROVE ?: Have everyone write down what they think can be improved, one idea per note. Post the notes, and group similar or duplicate ideas together. Discuss each theme as a team.

- Things that went wrong
- Things that need improvement

ACTIONS POINTS: From the previous points, select the most relevant AND the ones you can tackle in the next sprint.

From ONE to max THREE actions with responsible persons and timing