CSIC - Software Engineering

Project Guidelines

Irfan Hamid - Pablo Oliveira

ENST

19/01/2009

Outline

1. Team Composition

2. Phases of the project
   - Requirements and specifications
   - Design
   - Implementation and unit testing
   - Integration and functional testing

3. Delivery

Outline

1. Team Composition

2. Phases of the project
   - Requirements and specifications
   - Design
   - Implementation and unit testing
   - Integration and functional testing

3. Delivery

Team Composition

- Two teams of 3 persons, one team of 4 persons.
- Each team will have :
  - 1 leader
  - 2 or 3 technical member
- Leader's responsabilities :
  - Responsible for the organisation and communication within the team.
  - Establishes a project timeline with the help of the team, and assigns tasks to each team member.
  - Ensures that the project advancement is right on schedule.
  - Conception and implementation work.
- Technical member's responsabilities :
  - Helps the leader establishing schedule and work assignment.
  - Gives feedback to the leader and the rest of the team.
  - Conception and implementation work.

Technical member's specific roles

- The version manager :
  - Responsible of archiving the documents and the different code versions.
  - Responsible of setting up a VCS (version control system) that the team can use (svn, git, ...)
- The quality manager :
  - Ensures that all the team members write tests for their modules.
  - Ensures that all the tests pass, and checks for regressions.
  - Checks for problems during the integration of different modules.
  - He may set up an automatic testing framework to ease the task (Junit).
- (Only in the four member team) The documentalist :
  - Coordinates documentation writing among the team.

Outline

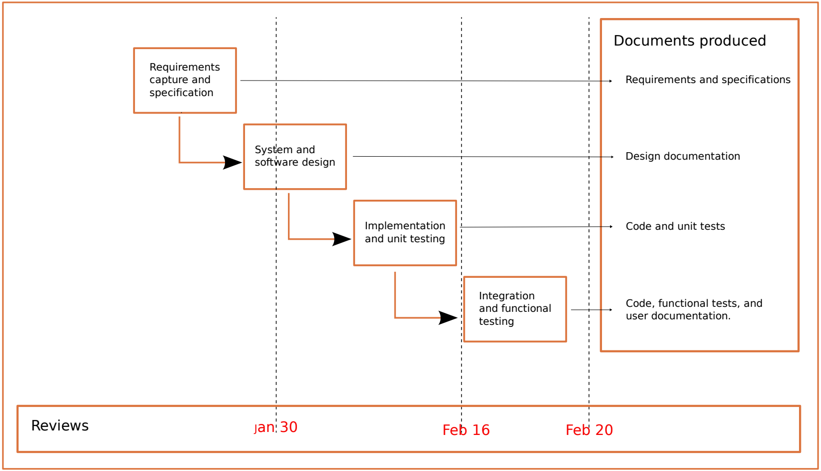
1 Team Composition

2 Phases of the project
- Requirements and specifications
- Design
- Implementation and unit testing
- Integration and functional testing

3 Delivery

Phases of the project

Because projects are simple, a waterfall model should work well.



Requirements capture and specification
System and software design
Implementation and unit testing
Integration and functional testing
Documents produced
Requirements and specifications
Design documentation
Code and unit tests
Code, functional tests, and user documentation.
Reviews
Jan 30
Feb 16
Feb 20

Iterations

- If you discover a problem in an upstream design phase :
  - Document the problem.
  - Produce an “Engineering Change Notice” to correct the upstream specification.

Requirements and specifications

- Produce a single document that states requirements and specifications.
- There is no imposed format for requirements and specifications (use what fits best).
- Requirements and Specifications must be numbered and cross-referenced. Dependances must be clearly stated.
- UML use-case diagrams are welcome to exemplify use-case scenarii.
- Do *not* start implementation before finishing specifications.

Design

- Think about the different modules that you will need in your project.
- Think about the algorithms you will need.
- Use UML when appropriate :
  - Produce a class diagram for the project.
  - If needed sequence diagram's can be used.
- Once you have decomposed your application in modules :
  - Define their interfaces
  - Delineate their responsibility.
- Write a design document that shows the module decomposition, the interfaces and the choice of algorithms. Join the class diagram of your application.

Implementation and unit testing

- Define a coding convention and stick to it.
- Code cleanly, factorize (do not duplicate code).
- Add *useful* comments to :
  - Each class
  - Each method
  - Each difficult to understand block of code
- Write unit tests while (or before) writing the code.
- Use a VCS !
- Communicate a lot in the team.

Integration and functional testing

- If your interfaces were well defined, integration should be smooth.
- Write functional tests for your entire application.
- Try corner cases.
- Do GUI tests by hand.
- Ideally, the person testing a functionality must not be the person who wrote it.

Outline

1 Team Composition

2 Phases of the project
- Requirements and specifications
- Design
- Implementation and unit testing
- Integration and functional testing

3 Delivery

Delivery

- Expected on time.
- Should include :
  - Commented Code
  - Tests
  - User Manual
  - README, that explains how to install and run the application (if needed state any external library requirements).
- Good Luck !