**Using Scrum and Agile in the Project**

**Scrum Implementation**

**Roles**

* Product owner – 1student
  + Owns the Product Backlog
  + Maintain the Product Backlog
  + Accept or reject the implementation of a user story at the end of the sprint
* Scrum master – 1 student
  + Make sure the team implements Scrums – Role, Ceremonies and Artifacts
  + Find solutions to the problems of the team
  + Makes sure everything is updated in Github. If not, follow up with the team to have the Scrum artifacts updated
* Developers
  + Communicate
  + Design
  + Code and refactor code
  + Test
  + Maintain code
  + Maintain artifacts
  + Produce demo video
  + Attend the required ceremonies
  + Update the required artifacts
  + etc
* Process coaches – Dr. Scharff
  + Provide weekly guidance to the Scrum team
* Testers – Team itself and students of the other teams
  + Submit bug reports
  + Summarize a report on the quality of the developed app
  + Determine if the content is relevant
* Content managers – 1 student
  + In charge of gathering the content
  + Content must not be copyrighted or sources have to be checked

**Ceremonies**

* Do 2 Scrums / week on Tuesday and Thursday
* Do the sprint planning *in one day* at the beginning of a sprint
  + Each tasks is estimated
  + The sprint goal is publicly available
  + The US are decomposed into tasks and assigned to the Scrum team
* Prepare and update the sprint demo&retrospective

**Artifacts**

* Produce the Sprint Goal at the beginning of a Sprint (1st day of the Sprint), i.e., the stories that will be implemented during the Sprint
* Tasks for the Sprint (1st day of the Sprint), i.e., the decompositions of the stories that will be implemented during the sprint
* Produce and maintain the Product Backlog
  + User stories with number of points (difficulty) and priority
  + New stories may be added to the Product Backlog during the Sprint
* Produce and maintain the Sprint Backlog
  + Tasks with estimations and corrections of the estimations during the project
* Scrum logs are available with the 3 questions
  + What did I accomplish since the last Scrum?
  + What will I do until the next Scrum?
  + What obstacles are impeding my progress?
* Bug reportsare available
* Produce a Burndown chart at the end of each sprint

**Agile Implementation**

* The Product Owner acts like an active customer
* Requirements are written as User Stories that are prioritized
* The evolution of the Product Backlog is tracked
* The project is divided into Sprints
* Team uses pair programming (at least once during the project)
* Team uses a coding standard
* Code is regularly refactored
* Team uses a code repository(Github) and git
* Testing is done early and often, and integrated throughout the project lifecycle (by developers and testers)

**Additional guidelines for Design**

* Produce an overall architecture of the app
* Produce a UI design using Cacoo or similar (optional)

**At the end of each sprint, the team produces a demo and a Sprint retrospective**

* Sprint goal
* Number of stories planned versus implemented
* Planned versus actual velocity
* Sprint retrospective written summary
  + What should the team start doing?
  + What should the team stop doing?
  + What should the team continue doing?
  + (in terms of process, communications etc)
* Status of the evolution of the Product Backlog
* Burndown chart for the sprint
* Statistics from GitHub
* Video demo of the application (YouTube link)
* Status of the software to be developed - summary.
* Icon of the app
* 3-5 screenshots
* Apk, link or equivalent

**Testers are testing at the end of each Sprint.**

* US have acceptance test for each user story
* Testers test at the end of each Sprintby using the documentation and artifacts. They also communicate with the developers.
* Testers test using emulators
* Testers test using real phones
* Testers test using web-based solutions
* Testers submits bugs in GitHub
* After testing, testers submit a testing report on their team (using the provided template)
* Testers post their testing report

**Use of a tool to support the software development process model**

* GitHub is used to support Scrum
* Requirements are in the tool (Product Backlog and User Stories)
* For each sprint:
  + Design is made available
  + Sprint Goal is made available
  + Scrums are made available
  + Sprint Retrospective is made available
  + The Code Convention of the team is made available
  + The Process Coach comments are made available
  + The testing report is made available
  + Bugs are submitted using the bug tracking system

**Communications**

* Each team uses different communication tools.