University of Wollongong

School of Computer Science and Software Engineering

CSCI311/MCS9311 Software Process Management

Autumn 2014

Group Project (45 marks)

TASKS

Your tasks are to:

1. Form and structure your group, allocating roles and responsibilities to your members;
2. Complete the development of the project described which should cover **all software development activities** including: estimation, planning, requirements elicitation and specification, designing, implementing and testing.
3. Produce a report detailing the group's work.

SUBMISSION

1. Mid-project deliverable (10 marks): in **Week 6 Tutorial**.
2. Final deliverable (30 marks): in **Week 12 Lecture**
   - Final report (hardcopy + softcopy)
   - Executable system + source code (softcopy)
3. Final presentation (5 marks): **Week 12 (in Lecture and Tutorial).**

GUIDLINES

1. The **mid-project deliverable** should be submitted as hardcopies covering the following:
   - A feasibility study of existing technologies that can be used and/or adapted for this project.
   - Detailed plan for the whole project
   - Software Requirement Specification (SRS)
   - Project progress up to Week 6
   - Member contribution up to Week 6 (with a signature from each member) – see details for this below.
2. The **final report** should cover at least the following:
   - Business cases
   - Project objectives and sub-objectives

- Risk analysis and measures
- Effort estimation
- A detailed project plan with progress
- Finale software requirements specification
- Architecture design
- A complete and detailed of the design including UML **use cases, sequence diagrams, class diagrams, state diagrams, data persistence and user-interface aspects.**
- Test plans, test cases and details of the testing phase.
- A user manual showing how your tool is used.
- Group meeting records
- Member contribution for the whole project (with each member’s signature)
    - In the cover page of your report, you need to indicate the contribution of each team member, and everyone in the team should sign the cover page. The individual contribution of each team member is assessed by all the other members (the scale is: “contributed”, “very little”, and “almost no contribution”). For a team member who has “contributed”, he/she will receive 100% of the group mark; for a team member who contributed “very little”, he/she will receive 50% of the team mark; for students who made “almost no contribution”, he/she will receive 0 marks for the entire group project. Your tutor/lecturer may make adjustment to this marking criterion based on practical situations.

Project Description

Important note: *This Project Description provides only the high-level goals of this project. The development team* ***is expected to elicit more detailed requirements as well as get feedback from "the client"*** *during the second half of each tutorial/lab.*

The development of software generates a huge amount of data in many forms such as source code, software requirement specifications, bug/issue reports, test cases, execution traces/logs, emails, discussions, and real-world user feedback, etc. For example, Mozilla Firefox had 800,000 bug reports, over 1,000 developers/contributors, hundreds of thousands of discussion emails, etc.

In today's software development, data plays a crucial role since hidden in the data is the information and insight about the quality of software, the dynamics of software development and the experience that software users receive. Such insightful information is extremely useful for software project management in learning from the past, understanding the present and planning for the future.

The aim of this project is to develop a tool that can be used as a dashboard for a given software project. The tool has three main goals:

- Importing and parsing existing raw data (e.g. bug/issue reports, test cases, execution traces/logs, emails, discussions, real-world user feedback, etc.) from a given large-scale open source project. **This data will be provided by the client**.

- Visualizing those data in a meaningful way to depict the current status of the project in various aspects (e.g. developers' activities, expertise and networks, user feedback, maintenance and evolution activities and progress, etc.).

- Allowing the user (e.g. project manager) to manipulate those data (e.g. create new tasks, requests, assigning tasks/requests, generating reports, etc.)