CS3500 Software Engineering

Assignment 2a: Software Project Scheduling

A.O'Riordan, 2020

This practical is part of the continuous assessment for CS3500. You will be marked on the answers you submit. Note that assignment 2 is in two parts. This is part a.

Submission details: You should submit both a PDF file of the Gantt chart and a CSV or Excel file of the data (generated from Teamgantt or similar tool). The due date is Monday 11th *December*.

For this assignment, you are to work as a **team of 3 people** (you can choose your own teams or I can assign you to a team if you don't have a team). Note only the team leader should submit the assignment. List in the body of the email all persons on your team. All team members will get the same marks.

Assignment 1a is on **project scheduling**. Part b next week is on **cost estimation**.

Teamgantt Software

The recommended software for 1a is the cloud-based teamgantt.com (this is free for one project). I am happy for you to use other project scheduling software if you can meet the requirements. If using teamgantt, one person should register for an account using UCC as the company name, then create a project and invite the other two people to your team. A PDF of the Gantt chart can be produced by choosing Print/Export option from Menu. A .csv file can be generated by choosing Export CVS from Menu. This can be viewed/edited with any spreadsheet software such as Excel.

Software Project Scheduling

Develop a *software project schedule* in the form of a *Gantt chart* for the software project described below. Three people (software engineers) are working on the project. Each student should assume the role of one of these engineers. Assume all three are working on the project full-time. The schedule should clearly show the *schedule of tasks (activities), task dependencies, milestones (end of phases), and the assignment of people (colour coded)*. You can make assumptions within the boundaries of the information given below.

A medium-sized **stock application** is to be developed using the Rational Unified Process (RUP) development method[1]. The work breakdown structure (WBS) consists of activities in various phases. Initial estimates of the cost of each activity have been produced (in terms of person-weeks) as part of the outline planning activity. If, for example, two people are allocated full-time to an activity that takes 4 person-weeks, it can be scheduled for 2 weeks. Note these are approximations; your schedule can vary from this.

[1] https://en.wikipedia.org/wiki/**Rational_Unified_Process**

Note that only two of the three engineers have expertise in requirements engineering and risk analysis. Likewise, only one of these two and the other person have expertise in implementation and test and integration and build.

Work breakdown structure (with estimates):

Inception phase

- Kick-off meetings (1 person-weeks)
- Outline planning (1 pw)
- Scope definition (1 pw)
- Risk analysis (1 pw)

Elaboration phase

- Requirements gathering (2 pw)
- Requirements specification (2 pw)
- Requirements validation (1 pw)
- Use-case modeling (2 pw)
- Test planning (1 pw)
- Prototype development (4 pw)

Construction phase iteration 1

- Architectural design (2 pw)
- Object-oriented analysis (3 pw)
- Design UI (3 pw)
- Design API for system (2 pw)
- Set up software configuration management (1 pw)

Construction phase iteration 2

- Detailed design of functions (6 pw)
- Unit testing (3 pw)
- Implementation and Test (6 pw)
- Integration and Build (2 pw)
- Test API (1 pw)
- Performance tuning (1 pw)

Transition Phase

- Deployment (1 pw)
- Acceptance tests (2 pw)
- Rollout (1 pw)
- Write user documentation (2 pw)

The end of each phase can be considered a milestone. There is no overlap between phases but activities within a phase can be done in parallel. Requirements gathering, requirements specification and requirements validation are sequential activities. Architectural design should be complete before object-oriented analysis begins. Detailed design and implementation can partially overlap. (Detailed design should begin at least a week earlier.) Integration and build likewise can start a week before the end of implementation.

end.