ECS506U Software Engineering
Group Project 2017

Guidelines, Schedule and Assessment

Version 1.0

23 December 2016

This document is a companion to the document *"Software Engineering Group Project 2017: Definition*."

Prof. Martin Neil

Table of contents

1 Introduction

The aim of the group project is to provide an opportunity to better understand the processes and methods of software engineering (as explained in the 'taught part' of the module) by putting them into practice. It also gives valuable experience of working in teams. You should expect the group project to be one of the most demanding parts of the undergraduate program.

As a group, the project tests your ability to analyse, design, build and test a software system. It tests your ability to make good decisions and recover from bad ones. It tests your determination to overcome difficulties, of which there will be many. It tests your ability to work within the constraints of limited resources (particularly time) and your ability to communicate technical concepts.

On average each group member should plan to spend approximately 120 hours total over the term working on the project. The rationale for this figure is that you are expected to spend 10 hours per week on the software engineering module over 12 weeks.

The detailed definition of the group project task for 2017 is contained in an accompanying document. The guidelines in this document describe:

- teamwork (section 2)
- labs and meetings (section 3)
- support from the teaching team (section 4)
- assessment, schedule and deliverables (section 5)
- assessing individual contributions (section 6)
- policy on collaborative work (section 7).

The course organizers are Prof Martin Neil and Dr. Mustafa Bozkurt. Teaching Assistants (TAs) will be allocated to each group to give technical advice to the group and to assist the course organizer.

2 Teamwork

The course organizer will assign each student to a group. Groups will be identified by an unique identifier. **Group allocation is final.** As in real life, you may have to work with people you don't know or like. As an individual, your task is to work with the rest of your group to produce a quality product. You must expect that things will go wrong. Group members will be struck down by plague and machines will explode. Teamwork is about managing resources, risks, and coping with difficulties.

Each group must select a project manager, whose responsibility will be to monitor progress of team members and report it to the TA assigned to the group, as well as submit checkpoint meeting minutes on a weekly basis (see Section 3).

ALL members of each group will be heavily involved in programming the system. In fact, **each student will be responsible for designing and programming one specific module of the system and responsible for integrating this into the rest of the system**. The module each group member is responsible for will be decided at random and is final. They alone are responsible for delivering this module in such a way that it can be executed on its own and capable of being integrated with all the other modules in the system. In this way a seamless and working integrated whole will be delivered to the customer.

We will be expecting each group to present a project plan to their TA for informal discussion. Failure to do so will inevitably impact the team's ability to deliver a system on time and of acceptable quality. You are expected to create specifications and designs, though these will not be formally assessed. You should know from Software Engineering theory that these are critical to the success of a shared engineering endeavor. They should also be regularly updated and evidence of how they change and develop should be minuted in the group checkpoint meeting minutes (see Section 3).

3 Labs and meetings

From week 2 each group is assigned a lab slot in the ITL when you should meet your group and work together. Attendance in the ITL is compulsory – a register will be taken and this will be used as evidence of individual contribution (see Section 6).

This slot is expected to be used for group development work, communication, interaction and keeping your assigned TA updated on the progress. **Outside of the assigned lab slot** you must meet as a group and work as individuals. Each week you must hold a ***checkpoint meeting*** (as a group you will, of course, require more meetings than this), for the purpose of recording progress, risks, issues and for planning. The group must keep a record of each checkpoint meeting **using the template on the QM-Plus module web page**. These checkpoint minutes must be recorded and submitted weekly as an item of coursework. The checkpoint meeting minutes provide information about each member's weekly contribution to the project, including a report on what they have done on their module. These minutes will be used to determine if any individuals are not fully contributing and so will be taken into account when allocating marks to individuals.

Failure to deposit checkpoint minutes by the deadline of Friday midnight at the end of each working week, from week 2, will result in a penalty that will be deducted from the whole group's final mark. If the checkpoint minutes do not record detailed plans, promises, risks and issues a proportion of the marks will be deducted.

If any member of a group is not contributing this MUST be recorded on the checkpoint minutes. If it is recorded elsewhere this will be ignored – it must be recorded on the checkpoint minutes *alone*.

Note that depending on timetabling arrangements there may be no ITL labs scheduled for week one but it remains imperative for your group to meet during this week to get started. We suggest you set up your project plan and check your group infrastructure then start documenting your designs during this week.

4 Support from the teaching team

Each group will be assigned to a TA who should be able to provide necessary support on technical issues concerning: Java programming, using NetBeans, UML, database, and configuration management systems. So, for any questions you have about these technical issues you should ask for help from the TAs during your assigned lab slot.

Outside of the labs, we provide the following support:

- *The software engineering course website*
- *The course web forum*. Questions posted to the forum are normally answered very quickly (often minutes rather than hours, so please make use of this resource). All queries that are not personal or group specific ***must be posted to the forum***, but before doing so you should check that the answer is not already on the course web pages (and that includes the lectures themselves, which are all available online). DO NOT email or telephone Prof. Neil, Dr. Bozkurt or Dr. Dementiev or any of the TAs with routine queries that cannot be answered using the forum or using information from the course webpages. This applies to clarifications regarding system requirements extracted from project definition.

While the above covers all technical and course related issues, personal issues (including any major problems with other group members) should initially be raised by emailing the Project Coordinator. ***You must include your group identifier in the email header and your own full real name in the body of the email. Messages that do not satisfy these criteria will not be answered.***

5 Assessment, Schedule and Deliverables

The final mark is out of 100. You must achieve a mark of at least 40 to pass the module.

We will be expecting to see project plan documentation, class diagrams, a list of ambiguities and how you decided to resolve them, during the labs. This documentation must be demonstrated to your TA on a regular basis. These materials will be needed because they are mandatory inclusions in your individual reports.

Ultimately you will be assessed on the final product and individual reports, as shown in Table 1.

Table 1 Marks and schedule

Deliverable	Week	Deliverable date	Delivery mechanism	Weighted marks per group member
Checkpoint meeting minutes	2-11	Each Friday midnight 23:59:59	Coursework submission system	10
Project repository submission and revision	2-12	Continuously	**All** group materials, source code, supporting libraries and databases, documents, test cases and checkpoint minutes	10
Release (of integrated system)	12	Final deadline Friday 31 March 2017 23:59:59	Coursework submission system	60
Test and Contribution report	12	Final deadline Friday 31 March 2017 23:59:59	Coursework submission system	20
Total				100

The various components of assessment are described in the subsections below.

All documents should be word-processed documents (containing, where appropriate, models generated by UML software). Marks will be deducted for poor formatting, grammar, layout and structure. ITL machines have Microsoft Office installed, and it is strongly recommended to use it for final formatting instead of the Open Office and alternatives.

The integrated system must be delivered via the coursework submission system.

5.1 Checkpoint Meeting Minutes

- The group must submit an electronic copy of the checkpoint meeting minutes via the coursework submission system, every week, starting week 2 and ending in week 11 (10 in total).

- The Project Coordinator, or his delegated nominee TA, will check each week that these are present, up-to-date, and accurate. The checkpoint minutes will form the documented basis for any disputes about individual contributions.

5.2 Project repository - Git

- A configuration management system called "Git" will be used for all storage, change monitoring, updating and delivery of all code, documents and test cases required for the project, including checkpoint minutes.
- Your software engineering group folder is under configuration control, and should be used by the group to store working material during the project life-cycle (as directed in the lectures). Details on the Git at QMUL can be found here:
  https://www.hpc.qmul.ac.uk/twiki/bin/view/HPC/GitHubEnterprise
- We will not assess or accept delivery of any item that is not available in Git, except for the individual report. Any group work submitted via the coursework submission system must also be traced on the Git.
- Since every group member will be coding a module this should be evident in the revision history in Git. This will be monitored by teaching staff and will be used to verify that students are meeting their commitments and submitting their own work. Typically, we expect to see only the name of the student responsible for a module listed against each revision to the module. Make sure to include descriptive comments when committing updates.
- At the top level there should be a folder called **master** and under this there will be a folder called **release**, another called **documents,** one called **working** a final one called **checkpoint**.
  - The **release** folder is where you must place the final integrated system installation files and setup. This must contain a final source code revision tag from the Git source revision used to build the release, with an executable with install instructions.
  - The **documents** folder is where you must place documentation, such as specifications and designs etc.
  - The **working** folder is where you should place your source code, libraries, configuration files and other system artefacts. You may wish to organize this in subdirectories reflecting your project structure. This working folder will be organized in sub directories matching the module names for your system.
  - The **checkpoint** folder is where you should place your project checkpoint meeting minutes.
  - You can create any folders and subfolders as you see fit so long as the above top level structure is respected at all times. Failure to do this will result in a penalty.
- A **scratch** folder should be created where each individual can store anything in any way they wish. Subdirectories should be created here under each group member's name. This is intended as a temporary ad-hoc work area. This folder should **not** be placed within the master folder.

- Failure to take Git seriously and use it will likely be terminal for you and your group. Using online systems like dropbox or facebook is not acceptable and we

will not recognise any submission that does not have a ***full*** audit trail, from creation to delivery, in Git.

- Tip: We expect students to start working on their own modules in their scratch area and as the project progresses and the central architecture and data model matures they will eventually all be working on the same integrated code in the working folder.

5.3 Release

The final code submission must be placed in the above specified folder in your group repository. We need the following:

- Files(s) necessary to run the code (normally a jar and/or executable). You must learn how to 'package up' your code in such a way that it can be executed by running a single file on any JVM-enabled machine. Failure to meet this basic requirement will almost certainly result in failure of the whole project.
- Files necessary to configure, connect and use any supporting data storage/retrieval components
- The above should be archived into a zip file and submitted via the coursework submission system. Note that there is a total file size limit on the submission which you should confirm with the ITL help desk prior to the deadline.
- Test your submission a machine different from the one you used for development. This will help avoid basic mistakes like a hard-coded full file path to the database file.

5.4 Test and contribution report

The test and contribution report is a confidential report that must be submitted as a coursework item. It should be no more than 10 pages and have no appendices.

It must have a front cover clearly identifying yourself and the group letter of the group you belong to. The report is expected to reflect three things:

- Your design: A summary description of the design of your module and how it integrates into the system as a whole.
- Your testing: A test plan along with achieved test results against each requirement for that module.
- Their contribution: A statement of the contribution made by each member of the group to project success

A template structure for the individual report will be issued and all reports must follow this format but, in outline, the individual report should contain (as a minimum):

- An explanation of the module you designed and built, including a description
- Describe this formally and provide design descriptions in UML. You must include key UML diagrams that help explain your contribution (for example a

class diagram that highlights the parts that you were responsible for). Marks will be deducted for not using UML and for not highlighting what methods and classes you produced.

- Test plan organized by module requirements describing the test case that should be exercised to demonstrate the requirement has been met. This should be detailed enough to allow someone else to successfully access the system and test your module.
- Test results documented against each requirement proving that you ran that test case and confirming the result (pass or fail).
- Brief description of how you felt the team worked together.
- A table listing each group member and assessment grades must be included in the individual report. The rows will be all the group members' names, as listed in the group membership spreadsheet (no alternative spellings and nicknames please!) except your own, and the column will list YOUR confidential assessment of the quality of contribution made by each member on a grade (E, M B and F).
- If all group members contributed satisfactorily and equally the grading should be a default M for each member. If you feel that the performance of a specific member of the group was below expectations, then grade them as a B and if above then an A. If they failed to make a significant contribution, then grade them F.
- In all cases where you grade a group member with any other grade than M a detailed justification must be given.
- In terms of grading we mandate the following interpretation, where expectations are as defined in this document (not your own!):

    1. E should be taken to mean contribution exceeded expectations
    2. M meets expectations
    3. B is below expectations
    4. F significant lack of contribution.

- Do not list nor grade yourself in your table.
- Do not share your grading or collude in your grading with any other members of your group. To do so will incur a penalty.

The majority of marks received for the test and contribution report will be for the test plan and results. This part of the document will be used to actually test your system and determine your final mark. Needless to say, a poor test plan will result in a very poor overall mark because system testers (us academics) will not independently prepare separate test cases or deduce how your system meets the requirement without the plan. In conclusion: failure to produce a test plan for a brilliant product will result in low marks.

6 Assessing individual contributions

By default, the individual grading will be used. However, for poorly performing group members the score will be lower than that awarded to others. Evidence for this comes from checkpoint minute meetings, the Git repository logs, attendance in the lab sessions, comments in individual reports, and personal assessment by members of the teaching team. We will differentiate between group members who refuse to contribute fully and those who are stopped by other group members from contributing fully. It is also possible for individual contributions to be rewarded with a greater score (this typically occurs in cases where groups are left short on numbers due to members leaving, or where there are more than one poorly contributing students).

7 Policy on Collaborative Work

You may discuss problems and approaches with anyone, but the analysis, design, coding and testing of each module should be the work of a single individual. Integrating these modules together into a delivered system is responsibility of the whole group.

We do not tolerate plagiarism**.** If your submitted work contains code, text, diagrams, code, ideas, data, methods or any other material that is copied (even from members of the same team) then it must be duly and fully acknowledged and referenced to their rightful source at the point of its use. Failure to do so is plagiarism. If you are not sure how to apply this definition in practice, ask your TA. The College's standard recommended penalty for plagiarism in one course is failure in the entire year's examinations; plagiarism in two courses will lead to expulsion from the College.

BEWARE: Plagiarism, the presenting of other people's work as your own, is a serious offence tantamount to theft. It will be severely penalized.

We are perfectly happy with students reusing or adapting components that they find on the web providing that full references are given in their individual report and full information about how the code adapts/interfaces to the reused code. Failure to do this (and we WILL discover it) will be counted as plagiarism and will result in a zero mark FOR ALL MEMBERS OF THE GROUP. Because the zero mark for plagiarism applies to all group members, the only way to avoid it for members who are aware of plagiarism but do not approve of it is to report the offence and the offenders before it takes hold. Ultimately the people who suffer most from plagiarism are the plagiarisers themselves who miss a valuable learning opportunity.