



**Risk Assessment and Decision
Analysis Research Group**

***DCS235 and AMSC235: Software Engineering
Group Project 2008-09***

Guidelines, Schedule and Assessment

Version 1.0

22 Sept 2008

This document is a companion to the document "*Software Engineering Group Project 2008-09: problem definition.*"

Table of contents

1	INTRODUCTION	3
2	TEAMWORK.....	4
3	GROUP SUPERVISION AND MEETINGS.....	5
4	ASSESSMENT AND DELIVERABLES	6
4.1	CHECKPOINT MEETINGS/MINUTES	7
4.2	PROJECT REPOSITORY	7
4.3	PROJECT PLAN/HIGH-LEVEL DESIGN	8
4.4	CODE INCREMENTS (WEEKS 13, 23).....	8
4.5	FINAL REPORT	9
4.6	POWERPOINT FILE.....	9
4.7	INDIVIDUAL REPORT (WEEK 24).....	9
4.8	REPORT ON 'PEER TESTING'	10
5	IMPORTANT INFORMATION ABOUT HANDING IN WRITTEN DELIVERABLES....	10
6	ASSESSING INDIVIDUAL CONTRIBUTIONS.....	11
7	POLICY ON COLLABORATIVE WORK	12
8	REFERENCES	13

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). Equally important, it tests your ability to communicate technical concepts.

On average each group member should plan to spend approximately 200 hours total over the two terms working on the project. The rationale for this figure is that you are expected to spend 10 hours per week including contact hours on the software engineering module over 24 weeks. Approximately 40 of the 240 hours are spent in lectures and non-project lab work.

The detailed definition of the group project task for 2008-09 is contained in an accompanying document [1]. The guidelines in this document describe:

- teamwork (section 2)
- deliverables and how they are assessed (section 3)
- supervision (section 4)
- schedule (section 5)
- final assessment (section 6)
- policy on collaborative work (section 7).

Please note that the details of schedule, deliverables, and assessment are only indicative at this stage and may be subject to change. All changes will be posted to the course web site and forum [2] (which you should visit *daily*).

2 Teamwork

The course organizer will assign each student to a group. Groups will be identified by their letter A, B, C, etc. 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 should elect a project leader.

While it is to be expected that some members of your group will be better writers, some better programmers and so on, you are not to divide the labour on these grounds alone. This project is to be a learning experience and each of you should get substantial amounts of practice in all the aspects of software engineering, not just the ones you are already proficient at. Each person must be involved in: writing some of the documents; editing documents; writing and testing code. You are not to turn one person into a full time project librarian just because you don't think their code is perfect. If one group member is a fantastic writer, by all means let them edit documents and help teach others how to write better. If another team member is a "wizard" programmer, have them be design leader, but they are not to develop the software for the whole system. Rather they should help the others improve their design and programming skills.

The allocation of work among group members has to be specified in the Group's project plan, which is a formal deliverable in week 7 (see Section 4). Moreover, this plan has to be regularly updated and evidence of its proper execution has to be minuted in the group checkpoint meetings (see Section 3). Hence, the assessors will be able to monitor the extent to which each group member is pulling their weight and will also intervene if group members are being stopped by other members from making a contribution.

3 Group supervision and meetings

Dr David Marquez will act as overall ‘manager’ for the DSC235 groups, while Peter Hearty will manage the AMSC235 groups. In the labs, the TA’s will provide further technical assistance for all groups. Every week after week 4 each group will meet their manager for 10 minutes. Weekly attendance in the ITL is compulsory. Each group is assigned a two-hour slot in the ITL on Tuesday mornings when you will each be guaranteed access to a PC. As a group you will, of course, require more meetings than this. In general you must plan at least one brief checkpoint meeting per week. The team should keep a record of each checkpoint meeting **using the template on the course web page**. These minutes must be recorded and placed in your group directory. Your manager will check each week that the minutes are there and up-to-date. **Attendance at, and records of, all meetings count in the assessment.** What this means is that:

- as an individual, you will lose marks for failing, without good reason, to attend meetings
- as a group you will lose marks for every week that you fail to produce accurate and timely checkpoint minute meetings

The manager will check that you are working properly as a group and that you are on target to meet deliverables. If there are major problems between group members that cannot be resolved by the manager then the problem will be referred to the course leader.

4 Assessment and Deliverables

Your project will be assessed continually, based on both formal deliverables and ongoing work, as shown in the Group Schedule in Table 1. The "Marks" column gives the marking scheme for each Deliverable or Action. The "Weighting" specifies the contribution to the final group mark, rounded to the nearest whole number. So if you receive 60 out of 100 for your project plan, you will be awarded $(60/100)*10 = 6$ marks towards the final group score.

Table 1 Group marks and schedule

Deliverable/Action	Week	Deliverable date	Hard copy report	Marks	Weight (DCS235)	Weight (AMSC235)
Checkpoint meetings/minutes	4-22	Each Tuesday 09.00		20	4	4
Maintain project repository	4-23			20	2	2
Project plan/high-level design	7	07/11/08, 12.00pm	Yes	100	10	10
Code Increment 1 (DCS235 students ONLY)	11	5/12/08, 12.00pm		100	15	
Revised plan and design	14	16/01/09, 12.00pm		100	5	5
Prof Fenton announces new game requirement	15					
Final code	22	13/03/09, 12.00pm		100	50	65
Powerpoint file	22	13/03/09, 12.00pm		10	2	2
Final report	23	20/03/09, 12.00pm	Yes	100	8	8
Peer testing report	24	27/03/09, 12.00pm	Yes	10	4	4
Total					100	100

Week 13 refers to the first week of the spring term.

Your final individual score for the course will be out of 200 (as it is a double-unit module) based on the components and weightings shown in Table 2.

Table 2 Individual marks and schedule

Deliverable/Action	Week	Deliver – able date	Written report	Marks	Weighting
Attend labs and consultant meetings	4-22	Each Tuesday		42	10
Individual report	24	27/03/09, 12.00pm	Yes	100	10
Individual group project score (this is the final % group score adjusted according to individual contribution as determined by checkpoint minutes and manager)				100	80
Exam				100	100

The various components of assessment are described in detail below. The formal written deliverables will be assessed using a) some general criteria covering presentation and style, and b) criteria specific to the content and for each particular deliverable. These formal deliverables should be word-processed documents (containing where appropriate models generated by NetBeans). In addition to the paper version of the deliverable, an electronic version must be placed in the group repository (failure to do this will result in lost marks).

The written deliverables will need to conform to the guidelines described in the reference [3]. Failure to conform to the guidelines will result in lost marks for your group.

All code increment deliverables must be packaged as jar files and placed in the group project directory.

4.1 Checkpoint Meetings/Minutes

- You must put an electronic copy of the minutes of your weekly checkpoint meetings in your project repository (see below).
- Your manager will check each week that these are present, up-to-date, and accurate and award one mark if they are. The checkpoint minutes will be the basis for the weekly discussion with your manager.

4.2 Project repository

- You will be set up with a group folder that has read and write permission for each project group member and read permission for each member of the software engineering teaching team.
- All deliverable must be placed in the group folder as follows:
 - *Project Plan.* The project plan deliverable must be placed in the folder called **Project Plan Delivered**.
 - *Final (group) report:* The final report must be placed in the folder called **Increment 2 Delivered Materials\Final Report**
 - *Powerpoint presentation:* The powerpoint presentation must be placed in the folder called **Increment 2 Delivered Materials\Presentation**
 - *Formal Code increment deliverables:* The increment 1 deliverable (DCS235 students only) must be placed in the folder **Increment 1 Delivered**. The final code deliverable must be placed in the folder **Increment 2 Delivered Materials\Executable Code**. In addition to formal code increment deliverables, you will be expected to demonstrate regular intermediate increments, especially during the second term. You
 - *Checkpoint minutes meetings.* These must be placed in the folder called **Checkpoint Minutes Meetings**

- *Peer Testing reports.* Your two peer testing reports must be placed in the folder called **Peer Testing Delivered Materials**
- In addition there will be folders for your working documents, including your code, designs and plans, testing, as well as your own meetings. These folders should be continually updated. For example,
 - The folder **Working Code and Designs** is where you should keep the latest version of your code and designs (i.e. your Netbeans project material)
 - The folder **HTML Documentation Of Code** should always contain the latest version of your code and design html documentation (which can be generated automatically from your NetBeans project). It is expected that you will generate this documentation on a weekly basis for inspection and assessment.
- Your manager will check the repository on a weekly basis and award one mark each week if the repository is in good order.

4.3 Project plan/high-level design

Details will be provided in the lectures and on the module forum. We will assess this document on the following particular criteria:

- Requirements specification
- Task plan and effort allocation for subsequent increments and deliverables
- Clear time and effort allocation amongst members
- High-level design class diagram

4.4 Code increments weeks 13 (DCS235 only), 22

Code increments must be placed in the above specified folders in your group repository. Write access to these folders will be withdrawn after the deadline. We need the following:

- Files(s) necessary to run the code (normally a jar and/or executable). You will have to learn how to 'package up' your code in such a way that it can be executed from a single file on any machine. Failure to meet this basic requirement will almost certainly result in failure of the whole project.
- zip file containing the project source code (preferably as a zipped Together project).
- zip file containing the generated html documentation

We will assess the code increments against the requirements and criteria specified in [1].

The final code increment will be assessed by:

- An independent TA
- The course leader
- An independent expert from within the Dept of Computer Science.
- Two of your peer groups

4.5 Final report

Details will be provided in the lectures and on the module forum. The report is expected to be your summary of the whole project (what you implemented, how you designed it, how you tested it etc.) You should include some key UML diagrams from your documentation in order to get across the main design, but we do not want to see lots of detailed diagrams because it is better to view these online. The report should contain:

- Details of the final design and code (including appropriate key class diagrams and other relevant UML diagrams).
- Full traceability of requirements (making it clear also what functionality has and has not been completed).
- Details of how it was tested.
- Details of how your code could be reused.

4.6 Powerpoint file

This should be about 5-10 slides and should highlight any special features/problems of your software that you would like to bring to the attention of those testing/using it (for example, if some functionality doesn't work properly or can only be accessed in a non-intuitive way).

4.7 Individual Report (week 24)

This should contain:

- Description of how the project went from your individual point of view (including what was good and bad for you, what you learnt, what you feel you should have learnt but did not, what you personally would do differently in a group project situation next time round).
- Description of how you felt the team worked together (including any particular problems that were encountered and how they might be avoided in a future project involving this group).
- A table, like the one in Table 3 The rows will be all the group members' names (with your own first) and the entries will be YOUR confidential assessment of the effort and quality of contribution made by each member on a grade (A, B, C, D, E, F) and the suggested weighting. If all group members contributed satisfactorily the weighting should be 100 for each member. If you feel that the performances of specific members of the group were either so bad or so good that it would be unfair to divide the group marks evenly then you should include a statement to that effect in the Comments section. However, the final decision will be based largely on the manager's assessment of the individual performances and the checkpoint minutes (see also Section 5 below).

Table 3 Personal judgement of team members' performance

	Effort (Grade A to F)	Quality of contribution (Grade A to F)	Suggested weighting (out of 100)	Comment
Glenn Hoddle.				
Ricky Villa				
Ossie Ardiles				
...				

The individual report should be no more than 3 pages.

Individual reports will be treated in confidence and must not be seen by other group members.

4.8 Report on 'peer testing'

In week 24 each group will be assigned the code of two other groups to test. We will provide a template for how you should test and how to report on the results of your testing (it will be mainly comments and certainly not 'marks').

5 Important information about handing in written deliverables

For campus students, the written reports must be handed in to the Computer Science departmental office. Documents should be stapled in one corner only, NOT bound or contained in a plastic folder.

Please use the front cover sheet provided by the administration staff and list the names of **all** group members. Each group member should then sign alongside his or her name. Finally the **consultant name and group letter should be written in large bold capitalised type on the FRONT COVER sheet (not an inside cover sheet).** Failure to do this will result in marks lost.

6 Assessing individual contributions

As shown in Table 2, each student is awarded a mark out of 100 that represents their own mark for the project. This is made up as follows:

- Mark for individual report (this has a 10% weighting)
- Mark for attendance at consultant and group meetings (this has a 10% weighting)
- Individual adjusted group mark (this has an 80% weighting)

The individual adjusted group mark is calculated as:

$$M/100 * \text{Overall group mark}$$

Where the overall group mark is calculated according to Table 1 and where M is the *individual scaling factor*.

By default the individual scaling factor is 100, since the default assumption is that all group members perform equally well. However, for poorly performing group members M may be lower than 100. In particular, the assessors will take account of the issues described in Section 2 and the evidence relating to it (evidence comes in the form of the various ongoing meetings and minutes as well as feedback on individual reports). By careful monitoring we will be able to identify which students are really not doing what is expected of them, and we will also differentiate those situations where students are being stopped by other group members from contributing fully.

An example of the kind of rescaling that can occur is as follows: Suppose a group's overall score is 70%. Suppose there is evidence that one member in that group refused to engage properly and generally provided only 50% of his/her required input. Then this student's M score will be 0.5 and so he/she will receive an individual adjusted group mark of 35%.

it is also possible for especially heroic individual contributions to be rewarded with an M score of greater than 100.

Agena Ltd and ITRS Ltd offer prizes to the best performing software engineering students on the group project. If one group's final mark is clearly the highest then typically each member of that group will win a cash prize (individual group members who have not performed satisfactorily will be excluded). The student with the best combined project/exam mark will also win a cash prize.

7 Policy on Collaborative Work

You may discuss problems and approaches with anyone, but the analysis, design, coding and testing should be your group's work. We do not tolerate plagiarism. If your submitted work contains text, diagrams, code, ideas, data, methods or any other material that is copied 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 consultant. 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.

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

We are perfectly happy with students reusing or adapting components that they find on the web providing that full references are given 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 0 mark FOR ALL MEMBERS OF THE GROUP. Because the 0 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.

8 References

- [1] Fenton NE, "Software Engineering Group Project 2008-09: Problem Definition", Queen Mary University, Sept 2008, *also available on course web site in pdf format.*
- [2] Course web site:

<https://intranet.dcs.qmul.ac.uk/courses/coursenotes/DCS235/>
- [3] Fenton NE, "Improving Your Technical Writing", Queen Mary University, Oct 2006, *also available on course web site in pdf format.*