

# CO510 Group Assessment – Overview

Version 2017.11.08.01

**Important: Please read all of this document before you commence any of the staged elements.**

## Introduction

The project is being conducted using a portfolio approach where you will work in groups to submit specified artefacts at specified times. This work will span the second half of the Autumn term and all of the Spring term.

There will be multiple stages covering the analysis, design, implementation and testing of a system. As is normal in software engineering activities, between stages there may be changes introduced to the overall requirements or delivery schedule, for instance, requiring re-factoring of work or rescheduling of plans made in earlier stages. For this reason, following stages will include requirements to document evidence of any such changes and appropriate testing, including regression testing of the functions delivered in the first iteration (to ensure that they still work).

**It is important to bear in mind that the coursework schedule and deliverables for this module remain tentative throughout because this is part of the Software Engineering experience where nothing is certain ... things can change as the development progresses. It is also important that you do not start the work for any stage until it has been announced by the module convener.**

## Group working – dealing with problems

**Important:** It is the responsibility of every member of a group to ensure that all other members of the group are familiar with the requirements of each stage of the assessment.

Try to nip potential problems in the bud early by creating a strong dynamic and follow-up quickly whenever a group member seems to have become detached. Where stronger action is required, the following 'card' system may be used by group members.

A system of yellow and red cards is available to deal with non-performing group members. If an individual is issued a yellow card then they will receive 50% of the group mark for the stage for which the card was issued. If they are issued a red card then they will receive 0%. The yellow card should be seen as a serious warning to the person concerned, and that their inaction is significantly affecting the work of their group. *A group member should seek to retrieve a yellow card at the next stage by an improvement in attendance, or attitude, or performance etc. It is their*

*responsibility to demonstrate this to the class supervisor (who will seek corroboration from the rest of the group). A red card can only be issued after a yellow card.*

Cards can only be issued by all active members of the group acting in agreement, and must be supported by evidence (for example, documented regular non-attendance at meetings; repeated failure to perform tasks allocated). A yellow card should only be issued for repeated failures over a period of time (say two or three weeks). Red cards should only be issued if these failures continue over one or more stages following issue of the yellow card.

*Recovery is not possible following the issuing of a red card and a mark of 0 will be awarded for the work of all stages from the point at which it was issued.*

A card is issued by emailing the person to whom it is being issued, with a copy to the module convener ([d.j.barnes@kent.ac.uk](mailto:d.j.barnes@kent.ac.uk)) and all other group members. The email should be sent by just one member of the group but all must be copied in to ensure that it is not being issued by on the basis of a single member's behalf. There may be some circumstances in which it is felt that a card is justified but that it is not possible to obtain the active support of all other members. In such circumstances, you should seek the advice of the module convener.

We sincerely hope that no red cards will be issued.

## **Work space and gitlab**

These will be allocated for each group. All submissions will be made using Moodle although the gitlab repositories will also be accessible for assessment.

## **Version control of documents**

You should establish practices within your group to maintain version control over all documents created by the group.

## **The proposed schedule in outline**

**Important:** All timescales and deadlines are potentially subject to change throughout the module. You should also be aware that changes might be introduced at any point in order to simulate the realities of real-world software engineering projects.

### **Weeks 7-11 Stage 1 (User stories and use cases)**

The project will be announced and groups will be allocated shortly after. A proposed system will be described from a user's point of view and you will be asked to analyse the description and requirements and formulate user stories and use cases from it.

### **Weeks 14-16 Stage 2 (Planning and class diagrams)**

Revisions will be made to the original scope of the work to make development more manageable within the available time. During this period, you will:

- Develop plans for how you will organise your group and schedule its effort over the coming term given the timetable of the assessment's stages;
- Develop plans for how you will use your group's gitlab space to coordinate your work and track issues as they arise;
- Develop plans for how you will ensure quality and consistency in everything the group does or produces;
- Analyse and refine the provided use cases;
- Start designing the system in UML from the revised use cases and other descriptions of the system.

It is envisaged that you will not start the coding during this period.

### **Weeks 16-18 Stage 3 (Baseline implementation)**

Baseline functionality of the system will be stipulated and you will complete any remaining elements of the design and start on the coding of the implementation. You will need to revise the plans made in the previous stage and document these revisions.

### **Weeks 18-20 Stage 4 (Review and implementation)**

In the light of your experiences with implementation in the previous stage, you will be expected to review and revise your plans for continued implementation of remaining functionality.

### **Weeks 20-23 Stage 5 (Review and implementation)**

Implementation continues, once again revising plans in the light of experience, if necessary. This period covers project week when you will be expected to devote a large amount of effort to this assessment. At the end of this stage you will submit the full project corpus of documentation and source code developed throughout the project. The corpus will include, but not necessarily be limited to:

1. An overview document introducing the reader to the corpus of material.
2. Design documentation.
3. Project management plans.
4. Project schedule.
5. A fully tested implementation of the required use cases. This includes all source code, tests and their results.
6. It is possible (probable) that not all use cases will have been implemented. Each group must submit their progress reports (planned against actual) to show how their development progressed, where they deviated from planned work and how they dealt with it.

## Matters you might like to consider before you start working, and as you progress

There are a number of questions you need to consider (and keep reconsidering) that will help you manage your work and develop a good product. Things such as:

1. What features are essential ("needs") and what might be described as "wants" and/or "likes"?
2. How usable will the system be? Easy, difficult? How will you actually quantify and demonstrate usability? Will there be a user guide? Have you someone independent who could try out and evaluate your system for you?
3. Will your system work and how do you prove it does, i.e., are you going to develop the system systematically or in an ad-hoc manner (plans, progress reports, etc.)? Will you have an audit trail of testing, correcting it, etc?
4. Who is going to the work on any particular aspect of the system? Just one or two people, or everyone? How will you manage collaborative effort on different aspects? How much will each person do and, again, will you have evidence to prove who did what? Will you have regular meetings so everyone is on track? Think about what might happen if anyone (or more) from your group was exchanged with members of other groups. How long would it take them to get up to speed (i.e., how well documented, structured and managed is your work)?
5. How will you ensure that the code you write can be understood by a newcomer to your group (design documents, coding standards, etc.)?

Preparation is essential. If you rush into coding without having considered all of the above, you may get a system that works, but not one that has been well engineered (which, in the long run usually leads to disaster).

The project is about Software Engineering, the **process** of developing a system that is usable and appropriate, and providing credible evidence that this has been achieved. It is this that you will get marks for!

(A comment from one of our Industry connections, taken from the film *Groundhog Day*. Whenever the main character wakes up in the morning he cannot remember anything about the day before. This may seem a bit extreme, but it illustrates the importance of keeping an account of your work and progress. The more ad-hoc your development the greater the risk of failing. Think about this and take care!)

David Barnes

Module convener

Date of this version: 2017.11.08