Module Code
COM00008I

BSc, BEng and MEng Degree Examinations 2019–20

DEPARTMENT OF COMPUTER SCIENCE

Software Engineering Project (SEPR)

Open Group Assessment

Module	Software Engineering Project (SEPR)
Year	2019/20
Assessment	2
Team	The Dicy Cat
Members	Michele Imbriani Daniel Yates Luke Taylor Isaac Albiston Martha Cartwright Riju De Sean Corrigan
Deliverable	Method Selection And Planning Update

Update to methodology used

URL: https://drive.google.com/file/d/17mzMUOYJLZijxw9pOtv8jCuqLC4iBdFd/view

We have decided to continue using the scrum methodology in our project. After completing Assessment 2, it is clear that a method with a high amount of adaptability is necessary, as tasks may take longer than originally anticipated. Using a more agile method will provide us with the flexibility we need to complete the next project in the three-week time frame. We will also try to keep the 1-week sprint length; however, understand that certain sprints may take a little more/less time.

The tools we will use for assessment 3 will go largely unchanged from assessment 2. GitHub was an excellent tool for keeping track of our progress, particularly as it meant that different people could work on different aspects of the project without disturbing each other's' code. We still plan to use Discord for online meetings and Google Docs to store our documentation. One change we do plan to make is to switch out as many online meetings with in person ones as we can. Although it can make it more difficult to plan a meeting, we found that it made a substantial difference being able to talk face to face; meetings were more productive and more organised. The recent technologies additions and their explanations and justifications can be found on the Method and Plans updated document highlighted in YELLOW. (see URL)

One change to mention which can be seen in the Gantt Chart is that for assessment 2 the implementation took much longer than originally anticipated- there are also a couple more tasks for implementation than before. The testing and documentation were pushed to the last week. For assessment 3, however, we split up the architecture so that there is a rough architecture before the implementation and the concrete architecture write up is after the implementation. We also pulled testing forward so that it happens at the same time I'm as the implementation as does the documentation. Assessment 4 had the fewest changes: we just brought the testing forwards again to give it more space. All these changed can be identified by being highlighted in LIGHT BLUE.

Updates to team management

Over the course of assessment 2, we ended up spending a long time on code implementation. This meant that the testing and documentation was given less time than we originally intended. To mitigate this problem in the future, we have planned to overlap the documentation and testing with the coding. This means someone will be writing the tests in conjunction with the others writing the code. This should not only give testing more time but will also ensure that the actual implementation will be written with testing in mind, a practice that will ensure that the developers structure their code to have fewer dependencies in between methods. These changes to the work breakdown have been reflected in the new plans, which have substantially more overlap than before. We have not yet decided who will be given what task for the remaining projects, as we found it easier to allot tasks as they arise rather than give everyone concrete roles.