ru43@kent.ac.uk, as2352@kent.ac.uk, rk410@kent.ac.uk, dd387@kent.ac.uk

Review outline

Review meetings will take place on a weekly basis, in order to ensure that documents and software are up to the desired standard. This review meeting will take place at the end of the week. We will use the meeting to identify what problems we have encountered while working on them during the week. The previous meetings minutes are used to make notes of what we need to work on and what we need to do moving forward on our documents and software project. These review notes will be used throughout the week and during the next review meeting.

If we find that during the review meeting we encountered many problems that we think relate to a plan or processes we are using, then we may adjust our plans to solve these issues. This would require us providing an amendment to the previous documents and plans we made. If a member of the group is not in the review meeting in which we change our plans, then we would notify them of said changes and try to setup a meeting for them to catch up with another member of the group to minimise misunderstandings and properly convey our problem and the solution.

Pair programming will be used for general programming, but also to ensure that documents are reviewed correctly. We will also use code review when we meet to ensure quality on code done separately. We will be using the most common type of pair programming, with one person being the driver and the other being the navigator. These two people will share a single computer. The driver will be responsible for using the keyboard and creating code, whilst the navigator will be revising and reviewing what is being typed. Ideally there will be a constant interaction between the two parties, allowing for a reduced workload along with other benefits such as transfer of knowledge and an increase in productivity. We will also employ a Strong-style pairing, where the driver and navigator can switch at any point if they have an idea or suggestion. We believe this will be ideal for our group as all of us are good programmers with a strong grasp of the theory behind the task. This will allow us to stay engaged and improve at both roles, whilst hopefully preventing some of the possible problems with pair programming, such as poor communication and lack of focus.