RAC 1

Dr. Winston Royce in his paper “Managing the Development of Large Software Systems”; which was published on August of 1970 in *Proceedings*, *IEEE WESCON*, a reprint from its original publication in *TRW*; describes the steps to software engineering through his nine years of personal experience managing large software developments for spacecraft mission planning, commanding and post-flight analysis. He explains that the barebone structure of software development encompasses two aspects: analysis and coding. Furthermore, Dr. Royce demonstrates that analysis is fundamental and that it needs to be implemented in the design, testing and coding of the software project. In addition, the analysis encompasses the planning and execution of projects,which ranges from the system requirements,possible software failures, physical limitations of the hardware and a myriad of other factors. For this reason, he is a strong advocate of extensive documentation, as it allows for meticulous and calculated increments in projects. Consequently, this modular increase allows external agents that were not associated with the original development to take over without wasting time or energy understanding its functionality. Another benefit from the extensive documentation is to efficiently correct code errors, as a testing team can spend its resources solving the issue instead of understanding what went wrong since the creator of the error can explain it in his own words.

Although Dr.Royce illustrates that analysis and coding are the minimum steps to create simple software projects, he acknowledges that larger and more complex projects require more steps to be successful. For instance, he suggests implementing a preliminary design step before the analysis to prevent software failures due to storage, timing, or data flux, Therefore, this is achieved by placing constraints on the program designer so that the software adheres to these limited resources. Nonetheless, there needs to be feedback between these two steps to allow the preliminary design to change if the software requires certain resources that were not allocated correctly. Albeit the steps outlined by Dr. Royce follow a waterfall managing approach, with sequential and discrete steps, it allows feedback between previous steps if something needs to be adjusted. Therefore, it can be argued that Dr. Royce's approach is a precursor to the agile development as these steps can be visited multiple times to create better software. However, it is not completely consistent with agile development, as the code is not protected to outside changes by sprints.

Speaking about large and complex software projects, it is apparent how developing software for mission planning, commanding and post-flight analysis requires meticulous steps with extensive documentation, and feedback between steps and teams. For instance, the Apollo 11 mission had several components that needed to work perfectly for the crew to stay alive. Therefore, implementing project developments strategies like the one outlined by Dr.Royce lead to their successful execution. Because, for a man to have walked on the moon, there had to be several teams working together to create a spacecraft that could withstand the harsh environment of outer space, while allowing astronauts to calculate daredevil trajectories and the command center on earth to monitor the process. Therefore, if the teams in NASA were able to put a man on the moon with the computing power of a calculator and following these managing steps, then everyone should follow this approach to create elegant and useful software.