Davie Truong

CMPS 115

dtruong8@ucsc.edu

RAC1-Managing the development of large software systems

Published on August 1970 in the Proceedings of IEEE Wescon, Dr. Winston Royce addresses his personal views about managing large software development. Using his experience with the development of software packages for spacecraft mission planning, commanding and post-flight analysis, he begins the article with the process of computer development. Stating the foundation of every program development has two steps, analysis and coding. The simplicity in the process paired with the creativity and work required for development result in a product that a consumer is happy to pay. However, with involvement of management in larger software systems, the process becomes convoluted. Many unnecessary steps get added that fail to contribute to the final product, resulting in tasks that developers would rather not implement and consumer not pay. Thus, Dr. Royce describes are more grandiose approach to software development.

With analysis, being the data of the project, and coding acting as foundation of the steps, the program development process receives: two levels of requirements analysis preceding, a program design step in between, and a testing step following the foundation steps. The purpose of the revised process is to ensure an “effective fallback position that tends to maximize the extent of early work that is salvageable and preserved.” Although Dr. Royce believes in this program development process known as the waterfall model, he is also aware the implementation is prone to failure, for which “the testing phase which occurs at the end of the development cycle is the first event for which timing, storage, input/output transfers, etc., are experienced as distinguished from analyzed.” This results in software that might not satisfy the external conditions, which result in a major redesign. Thus, Dr. Royce presents a revision with five additional features that eliminate most of the development risks.

Starting with program design, a new step is added between the software requirement and analysis steps. By having program design precede the analysis step, it “creates a relative vacuum of initial software requirements without any existing analysis, thus assuring that the “software will not fail because of storage, timing, and data flux reasons.” If a design is incorrect, it can be recognized at the early stage of development and the “iteration with requirements and preliminary design can be redone before final design, coding, and testing commences.”

With large software development having many intricate parts, Dr. Royce calls for extensive documentation. Acceptable written description forces the designer to take an unequivocal position and provide tangible evidence of completion.During the early phase of software development, the documentation is the specification and is the design.In the development process, documentation can concentrate personnel on the mistakes in the program and the manager can use operation-oriented personnel to operate the program and to do a better job.After the completion of the project, good documentation allows effective redesign, updating, and retrofitting in the field. In addition to documentation, stability of the product is important for the success of development. If the computer program is being developed for the first time, the version delivered to the customer should be the second version insofar. This grants precision during experimental tests in the questions of timing, storage, etc. Along with the three features, testing and customer involvement is an integral part to minimize developmental risks.

Dr. Royce’s approach is somewhat consistent with the agile approach because it offers the project development to be flexible in a way that if a problem arises, the team can work back to the previous strep. However, Dr. Royce’s approach is also linear and structured such that there isn’t much adaptive planning or evolutionary development.

The Apollo Computers article was not helpful in understanding the paper because it doesn’t delve into the topic of software development and how the management was organized for the operation.