Tu Lam

CS 361 / Dr. De Amicis

January 27th, 2021

# Propose Software Engineering

## (*A digital learning environment for schools*)

In the environment of digital learning for school, looking at the integration of features and type of services that can be provided to the student, I think that a way to implement this type of software project is through the process implementing all the features and going through the development step-by-step. By looking at the requirements, we can implement all those features in the high-level architectural model of a digital learning environment (iLearn) such as the application & the configuration services. These services including things like email, messaging, user storage, authentication, etc… and these features, as the developer team, we can implement these without changing the feature or the way the browser/app work. I decided to do this process of going step-by-step and having the previous stage approve before continuing is due to that the software does not need to change much in the school environment. As it stated that "designed for use in schools for students from 3 to 18 years of age" meaning that student start at kindergarten to high school uses the same system so all these functionalities won't need to change much as most student will use the same system.

From my own perspective, this will work well as a student, we do not need a lot of features and given these sets of features in the system is well enough for every student to use and making it not hard to learn. Even though it is stated that "also includes specific content, such as content about World War II and applications to view and annotate that content." This can be an optional requirement because not all from 3-18 years old student needed and that depends on the class and can be implement later.

This model process I use is similar to the Waterfall model, this model follows the process of looking at the requirement, design, implement, integrate, and operation processes. All these steps can be done by one step at a time and will need the approve of the previous step before moving on the next stage. And, after reaching the end of the process, we can go back to any points in the process and continue with the process to add new features or fix any problems that can occur after deployment.

With this we can see that the software is a fixed system that no new feature will not needed to implement and can follow the application and configuration service. Other type of services such as integrated, utility, and independent services can be change in the process as the

release of the software. Overall, the process is a linear process from the requirements to the maintenance and operation. This demonstrates that this system does not need to be implemented on the spot per requirement request, and the digital learning environment can serve best with this linear model of resemblance to the Waterfall model process.

Below would be a layout of what the process would look like in my perspective: