Mark Rosado

ELEC 3225-03

Professor Decker

6/14/2021

Assignment #3

Part 1:

1. Waterfall Model

For the CURSE Project if the waterfall model were to be used from start to finish the process would be linear and easy to follow but this provides risks and un-runnable code until the very end. The first step of this process would be identifying all the requirements and prep all other programs needed to complete this project for example, discussing with the team to see what kind of methods are required for each class and how much access each type of user should have when manipulating the roster, courses, or even other users. Next is the system design step which is how the data/program will flow both logically and physically. For example, creating pseudocode for the CURSE will be designed in this step, like having the user interface of the user interact their respected class through separate methods which will ultimately modify the database. The third step of this process is transferring the pseudocode created in the last step into actual programming languages, so in this step the database of the CURSE program will be written in SQL and the classes, methods and user interface will be created in either C++ or Python. The next step after is the testing stage which will have people outside of the development process run and deliver feedback towards the design. If any edits needed to be made, the process will return to the system design step. If not, the process will continue to the next step. Then, after the beta testing of the program, it will be released to the public and the process will enter the last phase, maintenance. In this phase, the team will receive feedback from active users on any bugs or improvements that could be made. The team will makes these changes and release a new version of the program with the corrections.

1. Incremental Development Model

The Incremental Development Model is the process of creating and testing a runnable portion of the total program until it is finished. Unlike the waterfall model there is less risk, implementing the code by pieces may provide smaller problems. Each increment will need the programmer to perform 4 steps to complete an increment. The first step of this process is called Analysis, this step will require the programmer and their team to discuss the needs and requirements of the this increment for example, in the CURSE program the very first increment would be creating the user interface with a basic parent class called user. The next step is the Design which is creating the pseudocode of the increment so in this case, the interface should access simple methods from the class just to show the programming logic works. Next, the coding step just as it says in the name is to code that smaller increment using here is C++ or Python. Lastly, the final step is to submit it for testing from outside sources to make sure there is no bugs. Repeat these four steps until the goal is achieved, in this CURSE program at least 2 more increments must be created, one to complete the classes and the other to setup the databases. Setting up the databases may be broken into two parts as they have different functionalities. However, this is up to the programmer.

1. Integrate and Configure Model

The integration and configuration model is the process of configurating existing software and implement it in once it meets the requirements for the goal of your program. This process can be used for any portion of this code that is missing. For example, if the GUI is not set up but the classes and the databases are, a solution would be to go on GitHub, or a site with open-source code, and search for one that has the foundation of a user interface. We would then take this code and edit it to fit the needs of the CURSE program. For this, we would need to modify this GUI to be able to take the desired input such as a menu with buttons for different user options. Overall, this method can be the coded the fastest but often results in less optimized code.

Example: <https://github.com/CITGuru/PyInquirer/blob/master/examples/confirm.py>

Graphical user interface, application

Description automatically generated

This python code is an example of how to create the GUI so I can use this to implement a GUI onto the CURSE program rather than having the terminal display a simpler interface. The first 10 lines of code would most likely be the same due to them being importing libraries however after that the variable names would have to change like “questions” to “options” and having the lines inside the brackets link to each method the class depending on what type of user is running the CURSE program. There may be more to incorporate into the GUI besides this but once the code has the databases implemented the next step would be to implement this code and see if a GUI can work.

SOURCES:

CITGuru. “CITGuru/PyInquirer.” *GitHub*, 8 July 2020, github.com/CITGuru/PyInquirer/blob/master/examples/confirm.py.

“Incremental Model in SDLC: Use, Advantage & Disadvantage.” *Guru99*, www.guru99.com/what-is-incremental-model-in-sdlc-advantages-disadvantages.html.

“Integrate, Configure, or Build? The Pros and Cons of Each Approach.” *Lineup Systems Homepage*, www.lineup.com/newsroom/integrate-configure-or-build-the-pros-and-cons-of-each-approach.

“SDLC - Waterfall Model.” *Tutorialspoint*, www.tutorialspoint.com/sdlc/sdlc\_waterfall\_model.htm.