Simeon Patton

HW 1 CS362

Spring 2021

Question 1

1. I would like to begin to imagine a phone or tablet application that would act as a HUD (head up display) in a vehicle. With all of the push to have everything within our lives smartly integrated, this app would act as a display to the driver for the pertinent information at any given time. For example, if the driver was speeding then the display would only show a warning message for the excess speed, otherwise the display would just show the current state of the drive itself or the vehicle. The display would be dynamic and based on what was happening at the time. Because there are already apps that focus on some of these features, however not all in a unified space, this would be a new software project with planned release date.
2. For this example, I am going to be choosing the Waterfall and Spiral methodologies to create this software program.
   1. **Waterfall** – This process would lend itself to functionality on this program because it requires very little input from the users. Since we most likely won’t have many users as it is an initial release, we will be able to produce the software in its entirety before releasing it to the customer base. Once the customers get the software in their hands, the waterfall method allows us to keep up with additional functionality and features with the maintenance phase of the process. With the waterfall method, we would have a clear understanding of the requirements going into the project based on the use of the software. Design, implementation and verification would be handled by the programmers and isn’t that dependent on a time frame as [once again] this would be an initial release of the software. Once the software if deployed, we can take user feedback for the next iteration of the program.
   2. **Spiral** – With this process, the program would undergo multiple iterations and builds before the release of the program. This methodology acts very much like the previous one except we are able to build upon itself at each iteration. First, we would still need to identify the requirements of the project, except this would happen at each iteration, so each build would have a new set of requirements for that specific build. This first step is also where we would determine and identify and new risks that would be associated with this build or addition to the program. Next, the developers would get to work on the actual coding of each specific build and its associated tests. Once the code is written for that build of the software, we move onto the planning and risk analysis of the next iteration. This allows us to stack functionality and features on top of an already working portion of the program up until the final release of the software.
3. With this particular software project, I believe that the waterfall method would be the best choice to begin with. This is mainly based on the fact that we specified that this was going to be a “first release” project. This one statement holds a few key factors that make the waterfall method a better fit. First, with a first release, we do not see much (if any) user input throughout the lifecycle process. This is mainly because the users do not have access to the software yet or simply do not know that it even exists. Second, when a company is creating a new program, most likely they want to keep the cost of the process to a minimum until they see that the software is going to perform in the landscape that it is intended for. The waterfall methodology addresses both of these concerns by being an extremely straightforward and inexpensive process, in comparison to the spiral methodology. The waterfall method also allows for some growth and change when it comes to the evolution of the program past the initial release to the public. If the end-users decide that they are looking for a different type of functionality and different features, then this process allows the addition and maintenance to the original program with updates.

Question 2

1. **Functional Requirements;**

* The table tennis AR software will display a playable surface to the users.
* The table tennis AR software will interact with user help motion controllers.

**Non-Functional Requirements;**

* The table tennis AR software will run on all compatible augmented reality platforms.
* The table tennis AR software will allow users to choose the color of their paddles.

1. **System requirements;**

* The system must include a through-put camera to display the user’s environment simultaneously with the software.
* The system must have motion tracking controllers that would allow the user to interact with the software.

**Software requirements;**

* The software must allow the user to enter their player’s name.
* The software must allow users to view the score of a game at any given point.

1. **User Stories**

- As a user I want to be able to log my high score in the memory of the game so that I am able to compete with my peers.

- As a user I want to be able to move the playable surface of the gameboard at any given time so that I am able to interact with a dynamic environment.

- As a user I want to be able to pause and return to the game at any point during the game time so that I am able to disconnect from the augmented reality space.

Question 3

1. **Scrum Master;**

A scrum master would take a specific user story/requirement from the product owner based on which sprint they are currently working on and guide their team to develop the software that focuses on the requirement at hand. The scrum master is responsible for the length of the scrum based on the requirement that the team is working on at that time. Working with the product owner to maximize the amount of work that is coming from the development team based on the current scrum while also holding the daily scrum meetings and managing the end of sprint review and retrospective, the scrum master has many roles to help get the software launched.

1. **Product Owner;**

Because the product owner is the only person that has the authority to arrange the requirements found within the product backlog, I would start by prioritizing the user stories for the development of the software. This priority is already accomplished when we are looking at the given user stories;

1. User login (all roles)
2. Enter search values (student and teacher)
3. Assign and update assignments (coordinator)

These user stories create the whole product backlog that the Product owner is responsible for managing throughout the timespan on each sprint. After the initial organization of the backlog, the product owner makes sure that the intended outcomes of the user stories are clearly understood by the development team. This ensures that there are very little issues when it comes time to actually code the software. As a product owner, I would have to make sure that there is a clear and concise line of communication between the users/stakeholders and the development team. It is the product owners’ job to translate what the users are needing into terms that are understood by the development team.

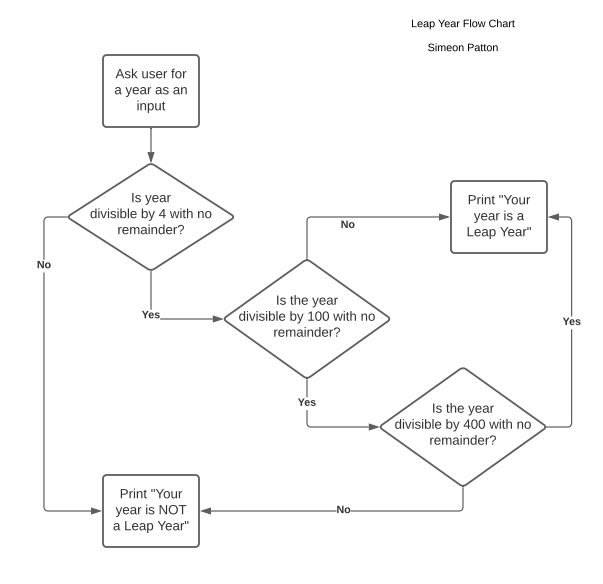
1. **Member of the development team;**

A member of a small development team needs to take the current requirement from the scrum master and focus on the creation of the product. This requirement needs to meet the needs of the product owner’s user stories that are found within the product backlog. A member of the development team is responsible for building the product correctly and with high quality so that the scrum master may move the development team and the project onto the next sprint. The development team needs to move through the assignments that are handed out by the scrum master to develop the software in its whole. The user stories that were recorded and organized by the product owner are all addressed one at a time by the development team until the product is ready for shipment to the end-user.

Question 4

Python Program

1. Flow Chart for Leap year Program



1. Please see the file named “Simeon\_Patton\_hm1.py” and the associated README file for the required program and instructions.