UFCF8S-30-2

Advanced Software Development 22sep_1

Part-2 Task 4: Agile Development Report

James Lear - Practical - CP1_03 Thursday 1430-1830

Date of Submission: 08/01/2023

Group 4

Members:

Rohaan Aslam: 21017718

Benjamin Hussey: 21022768

Yazeed Abu-Hummos: 21014295

Strategy Planning

The first step we took in the creation of the system was identifying the user requirements as outlined in the supplied brief, highlighting key requirements and information. We clarified any questions we had with the client, to ensure understanding of what the solution would look like. This allowed to us to outline the deliverables into a clear objective of: A user friendly GUI implementation of a secure OOP cinema booking system using an SQL based database allowing cinema staff to perform various work-related tasks with appropriate access control.

In order to finalize our plans for the system and user-type functionalities, we created a use case diagram to again show to the client and update with any changes they requested. This frequent communication with the client allowed us to strongly tailor the system to their needs and include all the features they wanted.

Now, having identified the problem and clarified any queries, we decided as a group that we would all take part in all the different aspects required to create the system, allowing us to all share our knowledge and benefit each other. An agile methodology we decided to adopt had been the XP agile practice. The XP practices allowed us to have a flexible & iterative approach which emphasised collaboration on the code by frequently testing each other's code, acquiring constant feedback and ensuring team satisfaction.

Continuous Team Iterations

One example of our agile approach would be starting the GUI (i.e., Login, Dashboard etc.) at the same time as working on the class diagram which meant we could get straight into coding functionalities rather than worry about aesthetics. Additionally, less time was wasted waiting on other team members to finish parts of the diagrams unrelated to the GUI.

Later on, we then revisited the GUI and revamped it to fit an MVC pattern. Another example would be the booking process, where data was fetched hard-coded in the view and then adapted to be fetched dynamically in the model reducing code. This constant iteration between planning and development was beneficial to us, as we could deliver features often to the client, meaning we could get feedback and then adapt and adjust our system to any changing requirements, as shown in the example.

In comparison, a waterfall methodology which is rigid and unidirectional, would have made it difficult to fit in our practices of switching between planning, development and testing leading to unnoticed errors and buggy code.

Team Coordination and Communication

Our team communication was based largely on face-to-face interactions. This gave us the ability to more easily clarify difficult aspects of both design and implementation such as: the model-view-controller pattern, any information stored in the database, the flow of data. We also had frequent discussions regarding changes of code which meant that all group members were knowledgeable on how the system was implemented as well as the reasoning behind the methods used.

A challenge faced had been online communication which was used to update other team members when commits to GitLab were made, or to query any uncertainties in work done or system design during the Christmas period. To mitigate this challenge, we established whenever each other would work on a part of the project to avoid accidental duplication of a page. Through constantly updating each other, our team peer reviewed & tested each other's work, ensuring that most bugs or errors could be ironed out quickly.

Simplicity

Another focus of our approach to working on the system was to follow the idea that “Simplicity is the art of maximising the work not done.”(Shore, 2007). This agile theory meant that we would sacrifice time on non-essential formal documentation and instead place focus on maximising the production of the system. The advantage of this is that the logs helped to serve as an alternative for documentation of the project.

We also followed a simplicity centred attitude towards the implementation of the system following the agile practice KISS (Keep it simple and stupid.). For example, we avoided the use of highly graphically styled widgets and focussed on displaying essential data especially in the early stages of development, as to avoid any potential harmful interactions with the styles.

References

Shore, J. (2007) *The Art of Agile Development* [online]. O'Reilly Media, Inc. [Accessed 05/01/2023].