

Team Project Individual Report - Team 12

Paul McHard - 2085227M

February 12, 2018

At the outset of the task our team made the decision to appoint both a Project Manager and a Scrum Manager, with the aim of adhering to the Scrum style of development and facilitating a well balanced distribution of both labour and responsibility. We felt this was crucially important working with a small team on a project of limited scope over an ultimately very brief time period; not only was the duration of the project only five weeks, the time within those weeks where it was possible either to meet or to work on the project was further limited. This was due to varying course choices and other commitments such as being part or full time students, as well as multiple team members having employment obligations beyond university. These differences were by no means roadblocks to progression of the project but were factors we felt were worth noting when structuring how we would progress.

My responsibility as Project Manager primarily involved promoting clear, consistent communication between the team. This involved maintaining momentum in the use of a group instant messaging system; making sure everyone knew where everyone else was in terms of progress, which was key for moving on to dependent modules of the program. I was also responsible for organising regular meetings. We made sure to have at least one full team meeting a week, as well as multiple meetings where at least a few members were present. These meetings were helpful as they were much easier and more flexible to organise, and could be kept easily to a focus of a particular task that needed addressing. Largely these involved only the people that had been working on code associated with that problem. These usually involved two or three members of the team and often occurred when we were trying to integrate two different people's work together. In these tasks it was not easy to anticipate where difficulty would arise, and so it was not an easy task to account for from a project management point of view. This was a key issue which could not be expressed in Scrum, and was on occasion quite time consuming. Therefore these small, flexible, sub-team meetings facilitated much faster solutions because having multiple people around a

computer made trying to fix strange errors which had no obvious solution much faster, and often it was the person or people who had not directly written the section or sections of code in question who could spot what was wrong most easily.

In terms of development work, I was principally responsible for creating and integrating the database for the system, developing the game methods for handling the end of rounds, such as the redistribution of cards and altering the user who won a round, and lastly handling how the game handled the addition of extra computer players beyond the initial one human one computer system.

Working on the database was a part of the project where I feel scrum worked very well; this section of the project was one which was quite easily developed in isolation, as well as being able to largely test and demonstrate its working status without integration into the main body of work. Similarly, but not to quite the same degree, addition of extra players to a game was a task that I handled and tested quite easily on the provision that we had, at that point within the command line, a fully working system for two players that we were confident about. The development of game methods for handling the end of rounds, however, was not as simple a task and involved a reasonable amount of concurrent programming. This style became quite frictionless and smooth but did take some getting used to, as coding alongside other people was not something that most of the team had experience in doing. I think this highlights the main drawback of using scrum for development in this project, which was simply that it was a novel system to use in practice for most of the team. We were largely brand new to actually using it and therefore weren't totally sure of how to use scrum properly. This is reflected in the differences between our team organisation document from the start of the project and our final group report. The former was created with the best intentions but as the project changed and development proceeded we moved away from the structure that had been therein laid out.

From my personal experience of the project, the end of round methods also ran into an issue of granularity, which is something the other members of the team found they ran into at times as well. In an effort to divide work as fairly as possible, in a few places we over-divided some sections, which could have possibly been produced more efficiently had they been compounded into a single task. This is an issue which is solved simply with experience in software project management. Similarly, as the program progressed we came to the realisation that some larger user stories within our program had been grossly underestimated in terms of ideal days required for completion. This led to the design change to implement a facade for the game object, which is reflected in our second sprint plan and was useful in gaining understanding about the disparity that can occur between estimate and actual completion times, especially when dealing with larger segments of work which have not been atomised.

One benefit of using scrum was that once we had created user stories and distributed out the work it was quite easy to make progress. The use of github facilitated sharing and incrementally changing and updating the code as different sections were completed. Taking an agile approach in which any member of the team who was able could take on work facilitated rapid progress and a quick turnover between tasks which were reliant on each other. Maintaining a high level of communication within the group was key to this as it meant we always knew where our priorities lay and where we needed to focus at a given point within a sprint.

The use of sprints allowed us to break up the work on another level and gave a more macroscopic view of how we were progressing within the timeframes we had set with the sprints. It was a useful system to employ because it provided a nice middle ground for mapping out a timeframe, between simply focussing on progress towards completing a module and progress towards completing the entire program. Creating a sprint plan, although a relatively simple one, allowed us to maintain a better reference of how the program so far was coming together at any given point and how that lined up to our broader time frame, particularly during the first sprint.

On a final note, while we employed many scrum principles and utilised its form of agile development within this project, we could not truly use an actual scrum system. This is highlighted within the points I have raised already within this report. Firstly, the program was too small and over too short a timeframe properly use more than two or three sprints. Secondly, as the program was being developed alongside multiple other courses and as individuals we each had other further commitments, there was no way to facilitate daily scrum stand-ups. Lastly, the program parameters were defined from the outset, and so there was no customer or customer proxy to allow for a true iterative design and development based on a continuous conversation around an evolving specification. Any iterative development within the project was thus of a limited scope and reflected either our varying design choices or changing views towards to sprint plan as we became more comfortable with using scrum. In conclusion scrum presented many benefits to the development of the project in terms of workload distribution and efficiency of development, but the limited scale of the program made it difficult to employ scrum to its full potential.