Team “H-Bomb”

Defiant Worlds Project Evaluation

From the start of the project, the group decided that the best life cycle and methodology to use would be an agile methodology. One of our main reasons for choosing this was that the way the project was set up allowed the group to get regular feedback from tutors which would allow us to refine our project using prototyping and incremental development. We felt that within the context of the project, customer collaboration (an important value of agile development 1) was a very important value.

During the initial project requirements analysis and design stage, project ideas were presented and then discussed in terms of feasibility and scope. When a project idea had been agreed upon, the group listed the high level requirements of the project in order to consider it as success and key functionality requirement that the project must have.

At this stage, whilst we feel that whilst the group effectively worked together to design an outcome, more could have been done to apply suitable scope to the project. Other than personal ideas of a preferred outcome, prioritising aspects of the project outcome could have saved the group time during development stage and would have given the project more structure. Whilst this could be seen as a contradiction to one of the principles behind the agile manifesto to “welcome changing requirements”[2], we as a group felt that since the core requirements of the project were very broad, that at this stage more prioritising could have been beneficial.

After we as a group had agreed upon the desired outcome of the project, and had organised a way in which to achieve this goal, we started the first prototype. Due to the setup of the project, weekly meetings had been organised, both for teams to communicate face-to-face and to have regular feedback from tutors who would be acting as the customer for the project. We decided to make use of this schedule and worked around a series of incrementing prototypes which would all have one week intervals. We agreed upon having a meeting after feedback to discuss any requirement changes, and then a 6-day sprint in order to develop and test the next prototype.

Initially, issues occurred through a lack of communication. Due to the fact that we were using an online repository, a lack of communication would results in conflicting changes being made to the repository at the same time. However this issue was addressed in an early meetings as being detrimental to the use of agile methodologies in the development, as being an effective, self-organising team is an agile principle [2].

Looking back at our experience, an alternative approach the group would have taken would be to value communication higher than we had done earlier in the project. Extreme programming, a highly valued agile technique [3] was not used during the project’s development as we placed a higher value of having individual areas of development to avoid complications. Use of this technique and other extreme programming techniques could have been beneficial to the group. Later on in the project our decision to have defined separations in feature development for group members resulted in time optimising code that could have been optimised during the sprint of its writing.

However, we believe that our chosen method of development was very effective in being able to produce prototypes in response to feedback in the short time we had allocated for project sprints. Each team member knew the area in which they were to work on at any given time, and our use of our online repository allowed for effective project management. The group learnt from earlier communication faults and conducted frequent stand up meetings, a key component in agile development to ensure that sprint goals where understood and any problems could be shared and discussed [4].

At the testing stages at each iteration, the group plan would be to have full unit testing of new developments, as well as having regression testing to ensure previous work had not been compromised by new development. Whilst testing was conducted regularly, thorough and documented testing was not done between increments. We as a group felt that unless the prototype was to be accepted as a final outcome, system wide testing outside of ensuring core functionality was not needed. Instead testing was done by ensuring that once a feature had been coded in, that a quick investigation into the program was done on possible areas that the programmer thought could possibly be affected. When bugs where found they were often fixed and then a test ran to ensure that particular bug had been fixed and little testing done on the effect of the fix, which is a key rule of Extreme Programming [5].

Nevertheless, the group still ensured acceptance testing was completed so that the prototype iteration was an effective response to customer feedback.

Upon reflection, more thorough testing could have been conducted between sprints. Due to the short time frame, testing was an aspect that was streamlined to allow for more development due to the scale of the project the group wanted to complete in the time frame given to complete it. An alternate approach would have been to construct tests before the next sprint, so that when features for that sprint had been completed, unit tests could have been conducted to examine all effects of the sprint on the project.

Overall our group considers the project a success. We set high bar for success in terms of what we wanted to develop and we were able to go above and beyond what was asked of the project. We were well organised and our use of the agile principles ensured that the group could work in an adaptive and motivated fashion.

However we were able to look back at how we conducted ourselves at each stage of development and identify areas in which we didn’t make use of several agile techniques that could have saved time and effort later on in development.

[[1]](#footnote-1)

References:

[1] – “[What is Agile Software Development?"](http://www.agilealliance.org/the-alliance/what-is-agile/). Agile Alliance. 8 June 2013. Retrieved 25 April 2015

[2] - Beck, Kent et al. (2001). ["Principles behind the Agile Manifesto"](http://www.agilemanifesto.org/principles.html). Agile Alliance. [Archived](http://web.archive.org/web/20100614043008/http:/www.agilemanifesto.org/principles.html) from the original on 14 June 2010. Retrieved 25 April 2015

[3] - Beck, Kent (1999). "Embracing Change with Extreme Programming". *Computer* volume **32** (issue 10): 70–77.

[4] - Vasiliauskas, Vidas (2014). ["Developing agile project task and team management practices"](http://www.eylean.com/Publications/DownloadPublication/3443705e-1697-4557-8327-ff8644fab40b?name=Whitepaper---Developing-agile-project-task-and-team-management-practices). Eylean.

[5] – Don Wells. (1999). *the Rules of Extreme Programming.* Available: http://www.extremeprogramming.org/rules.html. Last accessed 25th April 2015.

1. [↑](#footnote-ref-1)