Reflective Report

Project:

3.43 Weekly Planner App (Bulletproof Networks PTY Ltd)

Name:

Tristan Kells

Student ID:

1313715

Date:

02/11/2018

Supervisor:

Kenneth Johnson

Abstract

This report is a reflection of my personal experience of the final year Auckland University R&D project over semester one and two of 2018. We developed a high level team management tool for internal use at Auckland office of Bulletproof LTD. Bulletproof are the leading provider of cloud solutions in Australia and New Zealand. To deliver the our final solution, my team and I had to learn a new stack of technologies, design an innovative user interface and deliver on the educational requirements of a level 7 university paper. We utilised a selection of Agile Scrum processes to successfully complete the project, and were able to deliver a product beyond the original scope outlined to us. This report will attempt to critically reflect on the project, the development practices undertaken to complete it, my personal development and the effectiveness of our team.

Acknowledgements

I would like to firstly thanks my development team, Junha Yu and Dan Wood. They worked incredibly at all stage of the project to ensure we delivered something that would exceed the expectations of our stakeholders. They consistently went above and beyond what anybody could reasonably expect them to do, often giving up weekend to meet deadlines and ensure every meeting with our client included new features to show off. I’m grateful to them both for making my final year of university my best yet.

I would like to thanks our team mentor, Kenneth Johnson. He offered valuable feedback at all stages of the product and supported out learning throughout. He often asked us challenging questions that made us dig deeper and produce better work as a result. Me and my team appreciate all the time he has spent with us over what was undoubtedly a busy semester.

I would like to thank Krassie Petrova. As our team moderator, she gave us great insight at the pivotal stages of the project, the proposal and the status report.

I would like to thank the team at Bulletproof; James Burton, Craig Hurley, Grant Sweeney and Geethika Guruge.They were very supportive and gave us a lot of time at the beginning of the project to learn what we needed to deliver the project. Without their continued enthusiastic involvement, this project would have not been possible.

Lastly, I would like to thanks Charlie, Stephen and Karen for facilitating this project and giving us this fantastic opportunity to work alongside professional in industry. I have learned more this year than I ever thought I would, and am very grateful.

Table of contents

1. Introduction

For my third year at Auckland University of Technology studying a towards a Bachelor of Computer and Information Science, I completed a two semester long development project with two fellow students, Junha Yu and Dan Wood. Our project was to develop an application for the Auckland office of Bulletproof LTD. Bulletproof are top vendor of cloud solutions in Australia and New Zealand. They use both Azure and Amazon Web Services to deliver industry best solutions to clients. They used a whiteboard system to track the schedules of their consultants and provide an high-level overview of work being completed for clients over the week. This used a system that they had identified at other consultancy firms and wanted to see if a digital version of the system could be developed that would suit their use case. Our project was to create this digital system.

After consultation with Bulletproof and an analysis of our team skills, we decided the best solution would be to develop a mobile friendly, highly available web application. This application would allow us to improve on the features of the physical system, by allowing the storage of historical data, off-site updates and global sharing capabilities. We developed our solution using a LAMP web development stack and a Agile Scrum methodology. Development for the project happened over 24 weeks. From week 13 on, we delivered a working version of the application to the client every 3-4 weeks, so we could gather client feedback and deliver a high-quality product.

This report will be a reflection of my professional and personal learning over this year long development process. Firstly, I will evaluate our project. I will looks at how our our final deliverable compares with what my team originally promised at the project proposal. I will cover all significant changes made to the project scope and attempt to justify these changes when I feel it is necessary. I will also discuss my personal feelings on the significance of the project to the bulletproof team, and outline my concerns over long term product adoption.

The second section of this report will critical reflect on the practices and methods used throughout the development lifecycle. I will discuss our experience of using the popular Agile software development framework, Scrum. I will focuses on what practices I found most valuable to our software development and what practice we chose not to use. The following section will be about my personal development. This part of the report will focus on what I felt were the lessons I learned about myself and how I have developed a new set of skills as a result of partaking in this project. I will include what I have identified as important personal learning objectives as result my work, including some behavioral goals that I feel will make me a better developer and person going forward.

For the fourth part of the report, I will critically look at the performance of my team. I will discuss what I feel were our strengths and weakness, and how I plan to improve my ability to work in a team environment going forward. My focus here is not necessarily on the performance of my teammates, but instead about how I positively or negatively contributed to team collaboration over last year. The conclusion of the report will be a summary of my project experience.

2. Project Evaluation

A a high level, our final project aligns quite closely with what we planned to deliver in our project plan. We outlined and end up delivering a web application with an interface that matched bulletproof existing whiteboard consultant planning system. We leveraged the client's expertise to deliver on AWS. Our application has encrypted login. It is accessible by mobile device and tablet. We completed most high-level requirements, those we did not complete were no longer applicable with some of the changes we made to the product after client feedback. There are however a few noticeably difference between the initial product plan that I believe our worth mentioning.

We added an additional calendar view to the application, one that was not outlined in our project plan. At roughly the halfway point of the project, after a meeting at the Bulletproof office, Craig gave our team a tour of their offices. We saw there was another whiteboard system they used. It tracked consultant/client schedules over the course of 4 weeks, instead of one. We noticed this, and asked them if they would like us to add support for this information in our application as well. They requested we add this feature, but only as a stretch goal, as the information was not as important to them as the weekly view.

My team actually ended up completing version one of the monthly calendar far earlier than expected. Because we decided adding this feature would require a whole new application page, it was very easy feature to develop collaboratively as the work being done on it didn't interfere with work being completed on the main weekly calendar page. We often chose to develop feature because of how easily we could compartmentalize the work. Because we were all actively learning a lot about how to collaborate on a web development software project, we found this method helpful, however it often meant that higher priority work was sometime pushed aside. Although I personally feel this did not result in product lacking the necessary features, I imagine this would be something we should definitely not do once we are in industry. Instead, we should trust our teams/clients prioritization of tasks and take full advantage of the collaborative tools like github.

The original application administration plan was to have a manager account, who once logged in, would be responsible for adding consultant and client information to the application. Once a consultant has been added, they would be able to log onto the application using an account created by the the manager, and update their schedules. After consultation with our client we decided against this approach. Because the team at bulletproof was quiet small and close knit, for their use case it made more sense to give all members of the team complete access of the application, so everybody could work independently and with autonomy. As a team, we have discussed re-introducing concepts of administrative control if we continue development, which I believe we would need if we were to make the project into a commercial product.

Very early in the project, mobile/tablet optimization was given a high priority. Our clients imagined use cases often involved being able to retrieve and update consultant schedules on the go, using a mobile device. As the project developed and we investigated how mobile optimization might work, developing a feature-rich application that would support an uncompromising mobile experience looked to be slightly beyond the skill of our team, given the time frame. After discussion with our client, we adjusted the priority of mobile optimization and moved its implementation outside the scope of this project.

At our first client meeting, we talked to James about current tools his team used to track work and their time. He mentioned his team was actively trying to use Jira as a task management tool. Because of this, part of the original project plan was to make the application a Jira addon, which would enable us to leverage data from the Jira platform to populate our app. Ultimately, we and the client decided against this because of the poor login flow for applications that are part of the Jira ecosystem.

For some reason, we mentioned compatibility with Linux platforms as an important project milestone. This ended up not being an issue because as a web application, our software will run in any device with a web browser. When we wrote our project plan, our understanding of web development was limited, so we included some frivolous details, and exclude some vital ones.

Our client was pretty open about the fact that our project had a low priority to them. Bulletproof had two R&D projects this year. The other project was about creating AWS templates to help them roll out cloud solutions to their customers. In our first meeting with James, he mentioned that our project was the one they were less passionate about. Initially, I think we found this news kind of disheartening as a team. I remember coming out of the first meeting with bulletproof a bit disappointed that we seemed to have been given somebody second choice project. However, we realised as a learning opportunity we were in the exciting position of having a lot of freedom to do the project how wanted.

After the first semester break, when we had a lot more of the product delivered, I feel we were able to successfully get our client enthused about the project. We would get into meetings and show our progress, and complete small usability interview was we could collect information about how our client wanted the project to look, how they wanted certain features to function. As we progressed through the project, the client would give more and more feedback. This feedback was very valuable in helping us polish the project and indicated to me the increasing investment our client was making.

One of the major concerns for me throughout the project was making sure that what we made was something they would use. I could easily imagine a scenario where we put all this work into this project and delivered something they decided wasn't as useful as their existing whiteboard system. My gut feeling is this might be the case, as despite all of our feedback from the client, I still personally believe our application might not beat the raw usability of a tactile whiteboard system, for their use case. However, my team has discussed internally the possibility of making our application public and seeing if other teams out there working in a similar way could benefit and would be interested in using it.

3. Link Between Theory and Practice

Out project was completed using a version of the Agile Scrum methodology. We felt this would be the best option for our team, as we were familiar with it and felt we would benefit from the incremental development process and how it incorporated feedback. Client feedback was something we identified as very important early on because making something with great usability requires a lot of user feedback. By focussing on delivering the product in small increments, we were able to take advantage of this feedback to create a better final product. Although we took full advantage of incremental development, we did not use all the methods commonly associated with Scrum development.

When it comes to software development, Agile Scrum is the new standard which all other methods are held to. One of the key reason for this is how well the Agile mindset accommodate for uncertainty. Software engineering is a highly dynamic and fast moving field (Schmidt C., 2016). Technologies change and the desires of consumers in the information age evolve quickly. Scrum processes facilitate this uncertainty by enforcing a number of beneficial practices, that my team found highly valuable for our project.

Sprints in the Scrum framework are time-boxed events in which goal is to develop a potentially releasable product increment. Sprint can be any length of time shorter than a month or more than a week and they always include a sprint plan, sprint review, sprint retrospective, daily scrum / stand-up and all development work for a project. The power of this framework lies in the structure of these sprint. By forcing developers to break work into smaller increments of usable software, adjustments to the product and the team processes can made throughout the product life cycle. The other benefit of the sprints is enforcing exceptionally high communications standards.

My personal experience developing software and observing high performing team has indicated to me tha only 20-30% of software development is about coding. The rest of development, like all collaborative work, is communication. Scrum forces team of developers to communicate regularly and often, and this communication has obvious benefits for individual accountability and team cohesion. It makes sure that everybody is contributing towards a common goals, and that these goals are flexible to change and evolution as the project evolves.

As scrum master for our team, it was my responsibility that our team was actively participating in many of the practices of the Scrum methodology. Their were certain

Personal and professional development

Evaluation of own personal and professional growth against the originally expected skills and knowledge to be acquired in the project proposal/learning agreement.

One of the things that I have uncovered while studying is a passion for code. When starting my BCIS, I had a strong interested in the computer and the general belief that I was pursuing a career in a trending field. While on this 4-year long journey, I found code was something I really enjoyed. A session of development work can make me lose complete track of time as I'm fully engrossed in what I'm doing. This project, with a strong emphasis on development and web technology stacks was a fantastic learning opportunity. My interest in the topic paired with working for a team under a client a didn't want to disappoint meant that my coding skill has gotten exponentially better since starting this project. I have begun to really think about not only how to write clean, self-revealing and functional, but have come to a better understanding of why these things are important, through the real experience of collaboratively building software with my team.

Evaluation of the team roles and the effectiveness of the team members.

I feel that my close working relationship might have hindered my team's collaborative work. I considered one teammate a friend before the project started, and have definitely developed a friendly relationship with the other team member I had previously never met before. My relationship with both members of my team often meant that I was not willing to to strictly enforce meeting schedules, or pull people up for no completing their allocation of work for a sprint. At different times over the last few months, I struggled internally with varying amounts of resentment towards them, for the number of times I would arrive on time and then have to potentially wait hours before another member of the team showed up. Reflecting on it, I take a lot of the responsibility for not firmly calling them out on it, and setting some expectations about what was acceptable. They likely felt comfortable behaving this way because I failed to to call them out on it. Being too nice and accommodating of others is a common occurrence in my life, and the ability to set appropriate boundaries with personal relationships is something I will continue to work on. I was fortunate that both members of my team were highly capable, and although they did not always work to the schedules that we agreed upon, they would consistently produce high quality work.

Coming into the project with a good friend had another effect on our team which is worth noting. Me and Dan Wood would often work on the project together in the weekends. Sometimes we would make the effort to invite Junha, but if the collaborative work was spur of the moment, sometime he would not be included. The end result of this is that I don't feel like he was always adequately in the loop about the project. This was demonstrated often at sprint planning meetings, where Junha would only rarely be willing to take ownership of a user story. Which was a missed opportunity for us as a team, cause whenever Junha had work to do, he would do a great job at it, often developing at a speed well beyond me and Dan. While getting people to give up their weekends to work on a project should not be an expectation, a strategy we could of used to include him more would be to communicate a more regular schedule with updates on what me and Dan were discussing, or alternatively, taking a more managerial position directly assigning work to him, instead of waiting for him to take the initiative. Judging by his development skills, this would likely have resulted in a better product overall.

Conclusion

This R&D project has been the greatest learning experience I have had throughout university. As someone passionate about software development, I have at times resented the number of course I have taken that appears to only be partially related to this. While I understand the need for a tertiary institute to offer a breadth of material, so everybody has a chance to find their unique passion, once I decided I wanted to become a software developer, I really only wanted to sink my teeth into more development projects. This R&D project was a chance for me to do just that, and has really helped me soldiy my desire to become a software developer. I have learned valuable personal lessons by taking part in this project, as well as developed meaningful professional relationships.

References

Schmidt C.(2016) *Agile Software Development Teams : The impact of agile development on team performance*. Munich, Germany: Springer