

# Improving My Team Development Practices

COMP130 - CPD Report

1806868

May 3, 2019

## 1 Introduction

Since the last CPD I have improved in a few areas, I am now much more familiar with doing code reviews, and I have proven that I understand math by induction. I am also starting to get more confident with presentations. But there is still a lot that I need to improve on, to becoming a computing professional in the games industry with my own indie game studio.

There are several skills I need to improve to help me reach my goal. I still need to improve my time management skills to help prevent me from working overtime to get a task completed on time. I also feel it would be beneficial to learn some method to help resolve conflicts among the team and I would like to learn some advanced feature of Git, so I can better assist my team when things go wrong. I would also like to learn how to do more efficient sprint planning so that the sprint meeting doesn't go on for too long and I also need to improve how I give constructive feedback to try and prevent upsetting people.

## **2 Time Management**

During the group development project, I found it difficult to manage my time between supporting my team and completing my own tasks. Which often lead me to having to work overtime in order to complete tasks on schedule. As an aspiring game developer, I must develop better time management skills. I often find that I have to stop what I am doing to help another team member which then delays my work and I end up staying behind at the end of the day.

For me to improve on this further I will read ‘How to Manage Workplace Interruptions’ on Forbes [1]. I will also start to find out if anyone needs help during the daily stand-up to try and minimize interruptions throughout the day. I will also start to keep an interruption diary so I can work out what was important, what could have waited and what could have been delegated. I will do this throughout the first semester of our next group development project.

## **3 Resolving Team Conflict**

While working in a team we had several conflicts that made me feel uncomfortable and I found them hard to resolve. I sometimes find that it is really hard to find any middle ground that could help to resolve the issue. Since I want to own my own indie game studio, I must learn how to resolve conflicts effectively, so that the team can work as efficiently as possible.

To help me resolve conflict in the future I will read ‘Resolving Team Conflict’ [2] and I will go through the slides from Douglas Brown’s lecture. I could also try and sit down with the 2 parties in conflict to try and find a middle ground to resolve the issue. This should result in conflicts being resolved quicker and help keep the team focused and on track. I will ask other team members if they think I have dealt with the conflict logical and in an efficient manner. I will have completed this by the end of the next group

development project.

## **4 Advanced Git**

Throughout the last game development project, we had several issues that I'm sure could have been resolved using Git. One of which was all of our level went missing from the unreal engine although the files still existed. This resulted in the team having to create all the levels again since we did not know how to get them back. As an aspiring software engineer, I must learn how to use the advanced features of Git version control effectively and efficiently to help prevent having to redo all the work again.

To help prevent this from happening again I will read Pro Git [3] and complete all the challenges on gitbranching.js.org [4]. I already know a lot of the basic features of Git but I find that I still get stuck when there is a major issue that needs to be resolved and it is inefficient to do the work again. I have recently learned about the revert functionality in Git but I am still not that familiar with how its works. I aim to of finished reading Pro Git[3] and completed all the challenges of Git Branching [4] by the end of the summer break.

## **5 Sprint Planning**

Over the course of the last team project, we found it difficult to plan task for our game. At the start we over planned and created to many tasks that had no meaning and we could not measure their progress. This resulted in us having to scrap our scrum board and create a new one. To become part of a successful computing development team as a scrum master I must learn how to plan and organize a scrum board so that the whole team knows what they are doing and what to do next.

To help me improve this on the next project I will read 'Sprint Planning Meeting' [5] and '5 Common Planning Mistakes' [6] over the summer break. I already know about

using user stories but as a programmer I do not find them very helpful, although if I can improve on this it will benefit the next team that I work with. I will aim to put this into practice at the start of the next group project.

## **6 Constructive Feedback**

While working in teams there has been several situations where I must give constructive feedback. This is something I find challenging since I can be very blunt and brutally honest, which can end up upsetting the person who is receiving feedback. To be part of a successful team I must learn to give precise constructive feedback without being harsh, so that the team member receiving feedback has something that they can build upon and improve their work without being upset.

To help prevent me from upsetting people with harsh feedback I will read ‘Giving Constructive Feedback’ [7]. I am not very familiar with how to structure constructive feedback and it is something I need to learn so that the team can benefit. To see if I have achieved this, I will speak to the person receiving feedback several hours later to see how they felt about the feedback and if they feel it was useful, so I know I’m on the right path. I aim to of completed this by the end of the next group development project.

## **7 Conclusion**

If I can further improve my time management and get through the day with less interruptions I will become more productive which will result in having to do less overtime and if I can learn more of the advanced features of Git we won’t have to do the work again when things go horribly wrong. By Improving on my sprint planning techniques, it should make the team more productive since everyone will know what they have to do, when it needs to be done by and what’s next. By learning how to resolve conflicts efficiently and give better constructive feedback, it should help keep the team’s

morale high and give a better work output overall. If I'm going to own an indie game studio learning these skills are crucial to having happy and productive team.

## References

- [1] C. Scivicque, "How to manage (and minimize) workplace interruptions," [Online]. available: <https://www.forbes.com/sites/work-in-progress/2012/04/17/how-to-manage-and-minimize-workplace-interruptions/#4ee990516524>, [Accessed: 03-May-2019].
- [2] M. Tools, "Resolving team conflict," [Online]. available: [https://www.mindtools.com/pages/article/newTMM\\_79.htm](https://www.mindtools.com/pages/article/newTMM_79.htm), [Accessed: 03-May-2019].
- [3] S. Chacon and B. Straub, *Pro git*. Apress, 2014.
- [4] P. Cottle, "Git branching," [Online]. available: <https://learngitbranching.js.org/>, [Accessed: 03-May-2019].
- [5] M. G. Software, "Sprint planning meeting," [Online]. available: <https://www.mountaingoatsoftware.com/agile/scrum/meetings/sprint-planning-meeting>, [Accessed: 03-May-2019].
- [6] S. Porter, "5 common planning mistakes," [Online]. available: <https://www.scrum.org/resources/blog/5-common-planning-mistakes>, [Accessed: 03-May-2019].
- [7] M. Brounstein, "Giving constructive feedback," [Online]. available: <https://www.dummies.com/business/human-resources/employee-relations/giving-constructive-feedback/>, [Accessed: 03-May-2019].