

# Project Methodology

Alex Taylor - amt22@aber.ac.uk

February 16, 2017

Version 1.1 (Release)

## Contents

<b>1</b>	<b>Extreme Programming</b>	<b>2</b>
1.1	Justification . . . . .	2
<b>2</b>	<b>Use in Single Person Projects</b>	<b>3</b>
2.1	Practices Not Applicable to Project . . . . .	3
2.2	Practices to use on Project . . . . .	3
2.3	Process To Follow . . . . .	4
<b>3</b>	<b>Bibliography</b>	<b>5</b>

# 1 Extreme Programming

## 1.1 Justification

One of the core advantages of using an XP based approach is the ability to adapt to change. The second part of this project, implementing a method to stream slides from PowerPoint or a similar piece of software, is much more open to change than the first part where the scope is far more understood. If another process such as Feature Driven Development was used, it would work for the first part of the project as all the requirements and features are known but most likely would be a struggle to use in the second part. XP gives the flexibility needed to complete the second part of the project whilst also allowing the first part to be done with ease.

## 2 Use in Single Person Projects

One of the problems with XP is that a number of the practices are not applicable to a single person project, and a few others are not applicable within this project [1].

### 2.1 Practices Not Applicable to Project

- Pair Programming - Pair Programming would not be applicable because there is only one developer
- Collective Code Ownership - As above there is only one developer

### 2.2 Practices to use on Project

- Test Driven Development - Writing tests before coding any of the application logic will help to enhance both the design and also mean tests are actually written rather than left until the end and "hacked" in.
- Coding Standards - Keeping to strict coding standards will help ensure the code is good quality and easy to extend in the future as this project may well be.
- Small Releases - Small releases help enforce releases bit of working code regularly and will results in a better overall project.
- On Site Customer - This practice is somewhat applicable, there is a weekly meeting with the project tutor who is also the project customer. Feedback can be given during this meeting but also via email and other potential meetings.
- Merciless Refactoring - Refactoring is already encouraged by using TDD, but it can also be used at other points in the project to ensure the code is structured sensibly.
- Planning Game - This can be adapated to be done by one person at the start of the project, this is where all the stories will be written and then iterations and releases planned out.
- Continuous Integration - Depending on whether the project is hosted on-line, some online tools can be set up to run automated tests. This will be useful for running the tests written with TDD and ensuring the code is always working after an update.

Whilst not a practice, Class-Responsibility-Collaboration (CRC) cards can be used to enhance the design stage within each iteration.

### **2.3 Process To Follow**

During the research stage a process called Personal Extreme Programming was found [2] which gives a process script for working in an XP way but as a single developer. Whilst it does not specify practices to use, I have identified some above. PXP looks like a sensible way to develop and it will be the chosen methodology.

### 3 Bibliography

#### References

- [1] E. Akpata and K. Riha, “Can extreme programming be used by a lone programmer?” *Systems Integration*, p. 167, 2004.

Article describing how Extreme Programming can work for a single person project

- [2] R. Agarwal and D. Umphress, “Extreme programming for a single person team,” in *Proceedings of the 46th Annual Southeast Regional Conference on XX*. ACM, 2008, pp. 82–87.

Discussion of XP for a single person including what they call Personal Extreme Programming (PXP). Also describes the process for working in PXP.