

Android App for Local Taxi Bookings

| | |
|----------------------|-------------------------|
| Report Name | Methodology Outline |
| Author (User Id) | Rhys Evans (rhe24) |
| Supervisor (User Id) | Chris Loftus (cwl) |
| Module | CS39440 |
| Degree Scheme | G400 (Computer Science) |
| Date | February 10, 2019 |
| Revision | 1.0 |
| Status | Release |

1 Introduction

This document describes the Software Development Practices that will be applied during this project. The chosen methodology is an adapted version of Extreme Programming [1,2] to best fit an individual project. For this reason, many of the practices outlined in Kent Beck's 'Extreme Programming Explained' [2] are omitted from this document due to their focus on teams and team cohesion.

2 Practices

The following practices are all considered primary practices within Extreme Programming [2, Ch. 7]

2.1 Energized Work

Although mainly focused on software development teams to ensure a healthy workplace and work/life balance, this practice also suits individual projects very well. By following this practice, work is undertaken at the right time and location for the right length of time, without burning out. This requires that steps be taken to ensure a focused state can be entered to complete work without distractions.

2.2 Stories

Stories are a key practice in most agile methodologies, in this project user stories will be written for each desired functionality. Each story can be labeled as either 'essential' or 'nice to have', this ensures that work is prioritised correctly and essential aspects of the system can be built in a timely way. Stories will be written on physical, coloured cards and placed on a corkboard in the main work area of the project. They will also exist electronically on a Trello [?] board, this will allow progress to be shared with the project supervisor and allow work to be completed away from the main work station. There will also be a static document created with each high-level story written formally, this is for the project's final report.

2.3 Weekly Cycle

Weekly Cycles are synonymous with iterations in other agile methodologies. In this project weekly cycles will run from Saturday-Friday, meaning each cycle begins on a Saturday and ends on a Friday. The first day of the cycle will mainly consist of planning and choosing which stories to tackle; there will also be an overall reflection and a look at the project's current progress. The goal is that by the end of the cycle all stories have been realized and are now implemented, integrated and tested features within the system. The final day of the cycle will involve writing an entry into the project's weekly blog [3] reflecting on the week's work and documenting the goals for next week. By using a weekly cycle, software is constantly being produced, allowing for a regular opportunity to analyze and get feedback from the system.

2.4 Quarterly Cycle

Quarterly Cycles are synonymous with releases. Given the scale of this project (roughly 12 weeks), a quarterly cycle will consist of 3 weekly cycles. This means that the first quarterly cycle will pass just in time for the mid-project demonstration, thus ensuring there is at least a partly working system to be delivered. At the beginning of each quarterly cycle a plan will be drafted for which stories should be completed in that cycle. However, this plan can be

changed depending on the project's needs, the quarterly cycle plan will be re-visited briefly at the beginning of each weekly cycle.

2.5 Slack

The purpose of slack is to add low priority or non-essential stories to weekly and quarterly cycles. This allows tasks or stories to be discarded if time becomes an issue, thus accounting for inaccurate estimates.

2.6 Ten-Minute Build

The 10-minute build practice is intended to enforce a quick and easy build process, if a build process is long and arduous it is less likely to be done often. Therefore an easy, automatic build process will ensure that builds are done frequently, allowing for more feedback and less time between errors.

2.7 Continuous Integration

Continuous Integration supports the idea of a fast build process. By immediately testing and integrating code systems into the overall system, integration issues are caught much sooner and with less consequence. Continuous Integration also encourages and enforces frequent and reliable automatic testing.

2.8 Test-First

Test First is a prominent practice in many agile methodologies, it calls for tests to be written before any actual implementation. The following work-flow should be adhered to:

- Write failing test
- Run the failing test
- Develop code to pass the test
- Run the passing test
- Refactor

As was the case with the Ten-Minute build and Continuous Integration, a test-first approach ensures that issues within the code are spotted much sooner, hopefully resulting in more reliable and robust software.

2.9 Incremental Design / Refactoring

Incremental Design suggests that some spike work be done up front to understand how much effort is required and any issues that might arise. This approach will reduce risk and allow for a better estimation for a given story. This practice allows for informed design decisions to be made when necessary and made with the most current information available. The practice of incremental design also calls for the use of frequent refactoring to ensure correct design.

Initial Annotated Bibliography

- [1] K. Auer and R. Miller, *Extreme Programming Applied: Playing to Win*. Boston;London: Addison-Wesley, 2002, 0201616408.

This book is part of the XP Series, it is described as a pragmatic guide to getting started with Extreme Programming. This book will provide guidance on how to apply the important principles and practices of XP in this Project.

- [2] K. Beck and C. Andres, *Extreme Programming Explained: Embrace Change*, 2nd ed. Boston: Addison-Wesley, 2005, 0321278658.

This book provides an explanation of the practices and principles of Extreme Programming and gives an insight into the philosophy and motivations behind its creation. The book is authored by one of the original creators of the Extreme Programming methodology.

- [3] R. Evans, "Project blog," <http://blog.rhysevens.xyz/>, accessed February 2019.

This blog documents the progress of the project on a weekly basis.

- [4] Atlassian, "Trello," <https://trello.com/>, accessed February 2019.

Trello is a web based service that allows the creation of virtual boards, lists and cards to allow organization and prioritization in projects.