

# Student Lap Tracker

*Project Management Plan*

Patrick Oyarzun, Joe DeHart  
Vaishali Patel, Duke Meche, Robert Knott

## **ABSTRACT**

The following document describes the plan to manage the production of the product.

## **TABLE OF CONTENTS**

Introduction	2
Project Organization	2
Lifecycle Model Used	2-3
Risk Analysis	3
Hardware and Software Resource Requirements	3
Deliverables, Schedule	3-4
Monitoring, Reporting, and Controlling Mechanisms	4
Professional Standards	4
Evidence the Document has Been Placed Under Configuration Management	3

## **LIST OF FIGURES**

## **LIST OF TABLES**

### **I. INTRODUCTION**

- This plan articulates the major areas of concern for the lap tracking software project. Such areas include risk analysis, hardware and software requirements, schedules, and standards.
- The purpose of this product is to create a program capable of tracking the number of laps completed by students by scanning barcodes on their IDs as they pass. The scope of the product is to keep track of about 100 or so individual students in a single database, as well as have the capability to update any changes made to it from any devices linked to it.
- The plan is structured with organizational concerns as the first subjects, then with requirements, schedules, and management concerns, and finally with professional standards. The plan is also structured in such a way as to make it clear to any reader what is expected of group members and how they are to go about meeting said expectations.

### **II. PROJECT ORGANIZATION**

#### **Production Team**

Organization and roles of the team members are as follows:

- Patrick Oyarzun - Project Leader, Data Model (ERM, etc)
- Joseph DeHart - Query Planning

- Robert Knott - User Interface Implementation / Design
- Vaishali Patel - User Interface Implementation / Design
- Duke Meche - Client / Server Interaction

#### **Clients**

- Devin Olivier
- Eric Credeur

### **III. LIFECYCLE MODEL USED**

- Extreme Programming. Due to our hectic schedules, meeting with each other and with the client is difficult, so other models, like Scrum, are not optimal. By following the extreme programming model, we will be better able to produce reliable software. Due to the team's small size and ease of communication with the client, using the extreme programming model would produce better results than more process-focused methods. Also, with the use of GIT version-control, the constant building and rebuilding inherent in extreme programming becomes easier to manage, leading to a more coherent code and less misunderstanding with team members.

### **IV. RISK ANALYSIS**

- Specification Delays
  - Wait one week from initial contact; if no response is received, then send another email, or talk to Dr. Kumar.
- Absence of Group Member
  - Discuss reason for absence and attempt to find a solution. If no solution is possible, and future absences are expected, distribute work appropriately to make up for the missing person.
- Delivery Delay
  - Explain the situation to the client, apologize, and request an extension.
- Group Member Dropping Class
  - Adjust Group Member's roles to make up for the loss

### **V. HARDWARE AND SOFTWARE RESOURCE REQUIREMENTS**

- Hardware Requirements
  - PC with USB port
  - Barcode Scanner which acts a keyboard
- Software Requirements
  - Software Development Kit for Windows RT apps.

### **VI. DELIVERABLES, SCHEDULE**

1. First Meeting with Techneaux Tech (Sept. 23)
  - a. Discuss broad questions of the project, create a set of questions which need investigation.
2. Conference Call with Devin Olivier (Week of Sept. 29)

- a. Ask questions generated during (1). Verify current plan with her and set a time for (3).
3. Initial Prototype Delivery (2 weeks from (2))
  - a. Present first working implementation to client, preferably in person.
  - b. Discuss problems in the implementation
  - c. Discuss a “wishlist” of idealized features, if any, that the client wishes to have if we get ahead of schedule, with the understanding that possibly none of them will make it in before launch.
4. Polish the implementation (2 weeks from (3))
  - a. Fix problems discovered in (3.b)
  - b. Begin work on idealized features only after (4.a)
5. Meeting with Devin Olivier (Following (4))
  - a. Show and Tell the ‘mostly’ final project
  - b. Discuss final changes needed before launch
6. Make any necessary changes, further refine code
  - a. Fix problems discovered during (5)
  - b. Spend all extra time testing and validating the system
7. Final Delivery (~Nov. 2)
  - a. Demonstrate final product features to client, in person
  - b. Discuss how well the project goals have been met
  - c. Reflect on lessons learned

## **VII. MONITORING, REPORTING, AND CONTROLLING MECHANISMS**

- describe the management reports that should be produced, when these should be produced:
  - Risk Management
    - Need to identify all risks
    - Develop a course of action to minimize the impact of all
  - Budget Management
  - Time Management
  - Morale management
  - Progress reports
  - Client Update
  - Project plan
- and the project monitoring and control mechanisms used:
  - Communication methods
- include the rationale

## **VIII. PROFESSIONAL STANDARDS**

- Team members are expected to attend as many meetings as possible, but it is understood that likely no individual member will be able to attend all meetings.
  - If a member will not attend a meeting, he/she is expected to notify the group either at scheduling time, or at least 24 hours ahead of the meeting date.

- Team members are expected to comply with the university's policy on academic dishonesty. There is absolutely zero tolerance for plagiarism or any other violations.
- Team members are expected to meet deadlines set by the team and notify the other members if he/she will not meet the deadline so the work can be redistributed.
- Team members are expected to meet a certain level of quality in their work. It is understood that many of the tasks assigned are new skills that have never been used before, but a reasonable attempt at producing quality output is expected.
- By formalizing the above expectations, it is easier to maintain a clear set of guidelines for how the team interoperates.
- refer to Appendix A for more details

## **IX. EVIDENCE THE DOCUMENT HAS BEEN PLACED UNDER CONFIGURATION MANAGEMENT**

### **Appendix A.**

The following provides a professional standards guideline for the teams. This guideline may be tailored. The professional standards must be agreed upon by each member in the team.

Guideline:

On the first occurrence of unacceptable behavior, determine the circumstances involved, resolve the problem, and document the event in the meeting minutes.

On a second occurrence, notify the instructor of the problem. A meeting will be set up to evaluate the situation and resolve the problem.

On a third occurrence, again notify the instructor of the problem. A meeting will be set up to evaluate the situation and resolve the problem. At this point, the team will have the \*option\* of removing the team member. If removed, then the team member receives a pro-rated grade based on the number of weeks they have participated in the group.

Examples of unacceptable behavior may include not delivering on time, delivering poor quality work, missing team meetings, being unprepared for team meetings, disrespectful or rude behavior, etc. Reasons such as "too busy" or "I forgot", or "my dog ate my design model" are unacceptable.

Valid reasons that must be considered include those listed for obtaining an incomplete standing in a course (illness, death in the family, travel for business or academic reasons, etc.)