

LAKEHEAD UNIVERSITY  
FACULTY OF ENGINEERING



# Software Project Management Plan

*For the ChocAn Project*

**Instructor:** Dr. Ayman Diyab  
**Lab Instructor:** Mr. Mohammed AbuFoul

## North Light Software

Full name	Student ID
Joshua Whitlock	1272349
Thanh Vu Nguyen	1244225
Kenneth Shahi	1261283

Thunder Bay, 2025

# Contents

<b>1</b>	<b>Overview</b>	<b>2</b>
1.1	Project Summary . . . . .	2
1.1.1	Purpose, Scope, and Objectives . . . . .	2
1.1.2	Assumptions and Constraints . . . . .	2
1.1.3	Project Deliverables . . . . .	2
1.1.4	Schedule and Budget Summary . . . . .	2
1.1.5	Evolution of the Project Management Plan . . . . .	2
<b>2</b>	<b>Reference Materials</b>	<b>3</b>
<b>3</b>	<b>Definitions and Acronyms</b>	<b>4</b>
<b>4</b>	<b>Project Organization</b>	<b>5</b>
4.1	External Interfaces . . . . .	5
4.2	Internal Structure . . . . .	5
4.3	Roles and Responsibilities . . . . .	5
<b>5</b>	<b>Managerial Process Plans</b>	<b>6</b>
5.1	Start up Plan . . . . .	6
5.1.1	Estimation Plan . . . . .	6
5.1.2	Staffing Plan . . . . .	6
5.1.3	Resource Acquisition Plan . . . . .	6
5.1.4	Project Staff Training Plan . . . . .	6
5.2	Work Plan . . . . .	6
5.2.1	Work Activities and Schedule Allocation . . . . .	6
5.2.2	Resource Allocation . . . . .	6
5.2.3	Budget Allocation . . . . .	6
5.3	Control Plan . . . . .	7
5.4	Risk Management Plan . . . . .	7
5.5	Project Close out Plan . . . . .	7
<b>6</b>	<b>Technical Process Plans</b>	<b>8</b>
6.1	Process Model . . . . .	8
6.2	Methods Tools and Techniques . . . . .	8
6.3	Infrastructure Plan . . . . .	8
6.4	Product Acceptance Plan . . . . .	8
<b>7</b>	<b>Supporting Process Plans</b>	<b>9</b>
7.1	Configuration Management Plan . . . . .	9
7.2	Testing Plan . . . . .	9
7.3	Documentation Plan . . . . .	9
7.4	Quality Assurance Plan . . . . .	9
7.5	Reviews and Audits Plan . . . . .	9
7.6	Problem Resolution Plan . . . . .	9
7.7	Subcontractor Management Plan . . . . .	9
7.8	Process Improvement Plan . . . . .	9
<b>8</b>	<b>Additional Plans</b>	<b>10</b>
8.1	Security . . . . .	10
8.2	Training . . . . .	10
8.3	Maintenance . . . . .	10
	<b>References</b>	<b>11</b>

# 1 Overview

## 1.1 Project Summary

### 1.1.1 Purpose, Scope, and Objectives

The goal is to build ChocAn's Project, a software product that supports provider terminal and ChocAn database interaction. The product will let the client add, edit, and delete the needed records, run the required checks, and produce weekly reports and payment files. *North Light Software* will deliver a working product that meets all stated requirements, and ship the product to the client on time. *North Light Software* will produce clear reports and correct totals.

### 1.1.2 Assumptions and Constraints

Constraints include the following:

- \* We must meet the deadline and stay within the budget.
- \* The product must be reliable and easy to maintain.
- \* The design must allow new features later.
- \* The user interface must be simple.

### 1.1.3 Project Deliverables

*North Light Software* will deliver a working product, a user guide, technical notes, setup guide, test results, and acceptance record.

### 1.1.4 Schedule and Budget Summary

The workflow estimates are as follows: \* Requirements workflow – 3 weeks, three people, 12h. \* Analysis workflow – 1 week, three people 20h. \* Design workflow – 1 week, three people, 20h. \* Implementation workflow – 1 week, three people, 50h. \* Testing workflow – 1 week, three people, 30h. \* Total 10 weeks and 132h.

### 1.1.5 Evolution of the Project Management Plan

Any change to this plan must be approved by the engineering team, and recorded so the plan stays current.

## **2 Reference Materials**

All work will follow the textbook standards(Schach 2010) for coding documents, testing, and reviews.

### 3 Definitions and Acronyms

- **ChocAn:** The client company.
- **Client:** the sponsor or user of the system.
- **DataCenter:** the main system that stores records and runs reports.
- **EFT:** electronic funds transfer file for provider payments.
- **UI:** user interface.
- **DB:** database.

## **4 Project Organization**

### **4.1 External Interfaces**

All work on this project will be performed by Vu, Kenneth, or Joshua. AI usage will be cited. Questions that can't be solved internally will be escalated to Dr. A. Diyab via e-mail.

### **4.2 Internal Structure**

The engineering team consists of three Lakehead University Engineering students: Kenneth, Vu, and Josh.

### **4.3 Roles and Responsibilities**

Engineering communicates via e-mail if they have questions, keep records current on the git project, and escalate questions to Dr. Diyab via e-mail. Each member is responsible for the quality of their artifacts, and for verifying the quality of other's artifacts prior to submission.

## 5 Managerial Process Plans

### 5.1 Start up Plan

#### 5.1.1 Estimation Plan

Total time to hand-off is 7 weeks in total. Internal time cost is estimated to be 132 hours. These estimates are based on the class project deadline.

#### 5.1.2 Staffing Plan

The three engineering students are needed for all 10 weeks, first 5 weeks as analysts and designers, then as developers and testers.

#### 5.1.3 Resource Acquisition Plan

All tools and machines are available. The product will be delivered as source code, and the client will be expected to run it in their environment.

#### 5.1.4 Project Staff Training Plan

The engineers will be responsible for **first** reviewing the textbook and **then** consulting AI with any follow-up questions about their roles.

### 5.2 Work Plan

#### 5.2.1 Work Activities and Schedule Allocation

Week	Activity
Week 1-3	gather and inspect requirements, meet with the client, define group setup, and confirm scope.
Weeks 4	build analysis artifacts and inspect them, then get client approval.
Weeks 5	build design artifacts and inspect them.
Weeks 6-7	begin implementation by writing tests, then implement classes, integrate them, write docs, and do the final inspection.

#### 5.2.2 Resource Allocation

Each member works on assigned artifacts. Twice-weekly we will conduct a brief stand up before or after class to track progress and blockers. Scope changes will be confirmed with the client after the stand up. Documentation will be kept up to date in the docs/ folder of the project's git repository.

#### 5.2.3 Budget Allocation

Activity	Time-Cost
Requirements	12h
Analysis	20h
Design	20h
Implementation	50h
Testing	30h
Total	132h

### 5.3 Control Plan

Any change that affects time or budget must be approved by the engineering team and recorded. Each person tests another person's work to keep quality. Any major problems will be communicated with the engineering team via e-mail as soon as possible.

### 5.4 Risk Management Plan

There is currently no existing product to compare to, so we plan for strong testing. The ChocAn providers may have a limited technical background, so screens are to be kept simple, and frequently reviewed for simplicity. There may be possible design faults, so we review the designs often, and write tests early. There is a possibility of tablet or reader failure, so we keep spares on hand to give to providers to minimize downtime. Performance risk is low, due to small project size, but we will monitor response time. The choice of an interpreted language marginally increases this risk.

### 5.5 Project Close out Plan

*North Light Software* will submit our project, and all artifacts, on D2L on, or before, the due dates; or communicate with the client any delays in advance of the deadline. *North Light Software* will give the client the product, user guide, test results, and receive an acceptance note. *North Light Software* will then collect lessons learned.



## **6 Technical Process Plans**

### **6.1 Process Model**

We will use the Unified Process as described in the textbook(Schach 2010).

### **6.2 Methods Tools and Techniques**

Workflows will be performed as described in *The Unified Process*. The Product will be impletemed in Python. Diagrams will follow typical UML structure. Git and GitHub will be used in tandem for source control. GitHub Issues will be utilized to flag and close issues.

### **6.3 Infrastructure Plan**

Students will use a combination of their own laptops, their own operating systems, and collaborate through GitHub.

### **6.4 Product Acceptance Plan**

The client accepts the product after it is handed in on D2L and meets the requirements outlined in the assignment questions.

## **7 Supporting Process Plans**

### **7.1 Configuration Management Plan**

All files are stored in the git repository located here. We use branches for features, and pull requests for review.

### **7.2 Testing Plan**

Unit tests for classes integration tests for flows and product tests for full scenarios. Each person first tests their code and then tests a teammate code. The git repository owner runs final integration tests.

### **7.3 Documentation Plan**

We deliver the user any requested documentation or artifacts. All documents are to follow the same format as outlined in *The Unified Process*(Schach 2010).

### **7.4 Quality Assurance Plan**

Engineering should conduct peer reviews for all artifacts: when a team member finishes a feature, they request another team member to test it via e-mail. The git owner will be responsible for integration, and product testing.

### **7.5 Reviews and Audits Plan**

Not applicable at this time.

### **7.6 Problem Resolution Plan**

Any major problem is recorded as a GitHub Issue at once. The github owner reads it, assigns it, sets a due date, and tracks to closure.

### **7.7 Subcontractor Management Plan**

Not applicable to this project.

### **7.8 Process Improvement Plan**

We follow the team plan to improve repeatable practices and move toward higher maturity in the next term project. We will keep a record of our processes and lessons learned to iterate on our development process.

## **8 Additional Plans**

### **8.1 Security**

A password is needed to perform CRUD operations on the database. Providers login using their provider ID. All roles are limited to the needed actions.

### **8.2 Training**

Training will be given by Vu at delivery. One day is expected. Questions for the first year are supported at no cost.

### **8.3 Maintenance**

The team provides the software as-is, without warranty, with a full apache-2 license. Enhancements will use a separate agreement.

## References

Schach, Stephen R. 2010. *Object-Oriented and Classical Software Engineering*. 8th ed. McGraw-Hill.