**RISK MITIGATION, MONITORING, and MANAGEMENT PLAN**

**Michigan Treasury Local Government Data Parser**

**Senior Design I - Winter 2021**

**Table of Contents**

[**1.0 Introduction**](#_heading=h.msftlg2ekuxg) **3**

[1.1 Scope and intent of RMMM activities](#_heading=h.vqeasv29pufb) 3

[1.2 Risk management organizational role](#_heading=h.uspfd1ixq2y) 3

[**2.0 Project risks**](#_heading=h.ir7xwku2sib4) **3**

[2.1 Risk table](#_heading=h.o8nzyp3sa3oq) 3

[2.1.1 Description of risk](#_heading=h.o2le91oo31sy) 4

[2.1.2 Probability and impact for risk](#_heading=h.jpw7g29zgq9e) 4

[2.2 Risk refinement](#_heading=h.tehbuii61gw8) 6

[**3.0 Risk mitigation, monitoring, and management**](#_heading=h.56ezx0fbmlmr) **6**

[3.1 Risk mitigation for risk](#_heading=h.6a50zo8tx68m) 6

[3.2 Risk monitoring for risk](#_heading=h.hzqyvgasxk7h) 7

[3.3 Risk management for risk](#_heading=h.dah3l8gra2qg) 8

# 

# 1.0 Introduction

## 1.1 Scope and intent of RMMM activities

The goal is to make a piece of software that satisfies the client’s view and includes all the details they initially requested. Our team will work together with the client to meet their expectations and make the product run as smoothly as possible. While software is never perfect and may always contain some type of error, we will attempt to mitigate the number of errors. The purpose of this document is to calculate any risk the team believes we may encounter and devise strategies on how to deal with said risks while developing the product.

## 1.2 Risk management organizational role

The whole team is responsible for making sure all risks are calculated correctly and managed properly.

The whole team is responsible for developing the program and planning out the documents in order to stay on track.

The team is responsible for asking the client for all the requirements and information. Team is also responsible for staying connected with the client in order to see the progress on the program.

The professor is responsible for making sure to provide the team with honest feedback on our work in order for us to improve and succeed.

# 2.0 Project risks

## **2.1 Risk table**

## 

|  |  |  |  |
| --- | --- | --- | --- |
| **Category** | **Risks** | **Probability** | **Impact** |
| Developer Risk | Lack of experience/knowledge | 50% | 1 |
| Update Risk | Update program failure resulting in same or worse product | 25% | 1 |
| Quality Risk | Quality doesn’t meet client’s requirements | 15% | 1 |
| Client Risk | Client stops contacting us | 10% | 2 |
| Time Risk | Limited time to work on project | 15% | 2 |
| Tech Risk | The client doesn’t give us all the necessary requirements | 20% | 2 |
| Development Risk | The client doesn’t give us all the necessary requirements | 10% | 3 |
| Online Communication Risk | We cannot meet in person | 30% | 3 |
| COVID Risk | One of our team members gets COVID | 5% | 4 |

### **2.1.1 Description of risk**

**Developer Risk**

Our team comes from a few different major backgrounds so we don’t all have the same experience. Everyone in the group knows how to code, but each developer’s experience and coding proficiency varies. Some members, for example, may be more proficient in management than others while having less experience with coding, quality assurance, or documentation. The magnitude of code produced by each member is still undecided, as is the division of labor between coding and managing the project.

**Client Risk**

There exists a risk that the client may at times fail to acknowledge or interact with the team over the course of the project, due to any number of reasons. Previously, when the team submitted our product backlog, the client took a week to reply, indicating they may be quite busy. However, they have otherwise been reasonably involved with the project thus far.

**Time Risk**

The team may misjudge how long the program will take to develop and achieve its maximum potential. As we only have around 8 months to complete the project, the team must manage our time wisely.

**Development Risk**

The team has to ensure we have all the necessary resources required to develop the project. If the client doesn’t offer the team the adequate resources, we may not be able to produce the product originally requested. Thus far, the client has given the team the required resources. Still, in the future, there is a chance that the client may not give the team all the resources needed to complete the project.

**Quality Risk**

There is a risk that the program will not be top quality since the team has a limited amount of time to develop it. We also still have to figure out how the team will expand the program according to the client’s requirements. There is a risk that the client will not like the final product due to a potential lack of quality, or, at least, a product that doesn’t meet their expectations. It is incredibly important that the program is made correctly; it will handle vital government files and data that mustn't be manipulated or corrupted in the extraction process.

**Update Risk**

The team is updating the previous developer team’s project to meet the client’s requirements. There is a chance that our team fails to correctly update the software and instead damages it. We have to study and analyze what the previous team made in order to stay on track when updating the software.

**Tech Risk**

There is a chance that there already exists a similar program that the client can use, and the team is simply creating a different version of it. There is also the chance that the technology we have may not work as intended, affecting the team’s completion of the project.

**COVID Risk**

Due to the ongoing global pandemic, any number of our team members may fall ill and not be able to continue working on the project. This would cause the team to temporarily pause development and reevaluate our strategy, reassigning roles as needed in order to stay on track.

**Online Communication Risk**

The team has very limited in-person communication due to the ongoing pandemic, which may influence the resulting quality of our final product. We endeavor to communicate online through internet platforms, but it is perhaps more difficult to work with a team online than in-person.

### **2.1.2 Probability and impact for risk**

**Impact: 1 = Critical 2 = High 3 = Medium 4 = Low**

|  |  |  |  |
| --- | --- | --- | --- |
| **Category** | **Risks** | **Probability** | **Impact** |
| Developer Risk | Lack of experience/knowledge | 50% | 1 |
| Update Risk | Update program failure resulting in same or worse product | 25% | 1 |
| Quality Risk | Quality doesn’t meet client’s requirements | 15% | 1 |
| Client Risk | Client stops contacting the development team | 10% | 2 |
| Time Risk | Limited time to work on project | 15% | 2 |
| Tech Risk | The client doesn’t give the team all the necessary requirements | 20% | 2 |
| Development Risk | The client doesn’t give the team all the necessary requirements | 10% | 3 |
| Online Communication Risk | We cannot meet in person | 30% | 3 |
| COVID Risk | One of our team members contracts COVID-19 | 5% | 4 |

**Most Significant Risk:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Risk ID:** Developer Risk | **Date:** 02/24/2021 | **Probability:** 0.50 | **Impact:** Very High |
| **Description:** Our team comes from a few different major backgrounds so we don’t all have the same experience. Everyone in the group knows how to code, but each developer’s experience and coding proficiency varies. Some members, for example, may be more proficient in management than others while having less experience with coding, quality assurance, or documentation. The magnitude of code produced by each member is still undecided, as is the division of labor between coding and managing the project. | | | |
| **Refinement/Context:** Lack of experience/knowledge | | | |
| **Mitigation/Monitoring:** The team can avoid much of this risk by doing research about the project aspects and which tools are required to develop the software. We can then strive to learn how to use the tools in order to form the program to the client’s standards. | | | |
| **Management/Contingency:** The plan is if the team keeps getting stuck on part of the code we must seek help in order to solve the problem quickly and stay on track. | | **Trigger:** Stuck on developing phase | |
| **Current Status:** High Risk | | **Current Status Date:** 02/24/2021 | |
| **Originator:** Adham | | **Assigned By:** Adham | |

## **2.2 Risk refinement**

High probability/high impact risks are refined using the CTC approach.

Given that we may lack some experience and knowledge about the current job, there is a concern we will not be able to fully realize the client’s vision for the product.

Given that we have limited in-person communication, there is a concern that the final product will not fully meet the client’s expectations.

Given that we may not update the program correctly, there is a concern that the final product may or may not be in an adequate state for the client’s needs.

# 3.0 Risk mitigation, monitoring, and management

## 3.1 Risk mitigation for risk

**Developer Risk**

The team can avoid much of this risk by doing research about the project aspects and which tools are required to develop the software. We can then strive to learn how to use the tools in order to form the program to the client’s standards.

**Client Risk**

The team can avoid this risk by contacting the client weekly to provide updates about project progress. We must keep the client informed on our development progress and request meetings whenever we require more information about a certain project aspect.

**Time Risk**

In order to avoid this risk, the team needs good time management so we stay on track with the due date of each project section. Using COCOMO, we can estimate the time required to complete the project.

**Development Risk**

The team can avoid this risk by strengthening our understanding of the project requirements and preparing for any difficulties. In order to do this, we must observe the client’s needs and not create something different.

**Quality Risk**

In order to avoid this risk, the team can make a prototype and present it to the client in order to obtain feedback on the direction of the project. We must also finish all planning documents in order to stay on schedule and know exactly what everyone is doing to deliver a quality product on time.

**Update Risk**

In order to avoid this risk, the team has to analyze the previous team’s program and understand how it was developed. We then have to research and learn how to update the software in order to meet the client’s standards. If we don’t discover how to update previously-developed work, we may have to recreate the program from scratch.

**Tech Risk**

In order to avoid this risk, the team must search the web to see if there are similar programs already developed.

**COVID Risk**

The team can avoid this risk by taking the necessary safety measurements, including always wearing a mask when outside in public and keeping at least a 6 feet distance between us and others. We must be careful about our health and make sure to stay home if we feel any symptoms so we don’t affect other people.

**Online Communication Risk**

In order to avoid this risk, the team must capitalize on any in-person meetings if we have any while also staying in connection with each other through online communication. We must try our best with communicating with the rest of the team to work on this project through programs like Discord, Zoom, and Google Drive.

## **3.2 Risk monitoring for risk**

**Developer Risk**

The condition to monitor to determine whether this risk is becoming more or less likely is how often the team becomes stuck when developing the project. If we keep getting stuck on part of the code, we must seek help in order to solve the problem quickly and stay on track.

**Client Risk**

The condition to monitor to determine whether this risk is becoming more or less likely is how often the team contacts the client. We must keep the client updated and attend meetings with them whenever necessary.

**Time Risk**

The condition to monitor to determine whether this risk is becoming more or less likely how much the team falls behind schedule. If we remain on schedule or become ahead of schedule, we will most likely not experience this risk. However, in the event the team falls behind, we will certainly face this risk.

**Development Risk**

The condition to monitor to determine whether this risk is becoming more or less likely is the team’s understanding of the project and if we have the necessary resources for completion. If the client has provided the required resources and we do enough research, we should be able to avoid much of this risk.

**Quality Risk**

The condition to monitor to determine whether this risk is becoming more or less likely if the team experiences bugs or errors in the program. If the program is of a good quality, then there shouldn’t be many bugs and the risk should be minimal.

**Update Risk**

The condition to monitor to determine whether this risk is becoming more or less likely is how well the team understands the previous team’s project. If we have a good understanding of the project, we should be able to use what the previous team built in order to make a successful final product. However, if we don’t understand it well then it will affect our updating of the program.

**Tech Risk**

The condition to monitor to determine whether this risk is becoming more or less likely is if other programs already exist that are similar to the one we are developing. If there are, then the risk will decrease because we can use something as a reference to what we are building. If there isn’t a similar program then the risk may increase as we will have to develop something entirely new.

**COVID Risk**

The condition to monitor to determine whether this risk is becoming more or less likely depending on how each team member takes care of themselves. We have to prepare for the worst and always take safety precautions to avoid contracting COVID-19.

**Online Communication Risk**

The condition to monitor to determine whether this risk is becoming more or less likely relates to how many times we are able to meet in person. If the pandemic status changes and we have more in-person meetings, then the project will be easier to develop.

## 3.3 Risk management for risk

**Developer Risk**

The plan is if the team keeps getting stuck on part of the code we must seek help in order to solve the problem quickly and stay on track.

**Client Risk**

The plan is to obtain all the necessary information from the client as soon as possible and ensure the team remains on track in case we lose touch with the client.

**Time Risk**

The plan is to make time management tables and schedules to stay on track.

**Development Risk**

The plan is to have all the resources required in order to develop the program to the best of our ability and not risk producing a useless product.

**Quality Risk**

The plan is to test the program and code throughout development and check for errors and bugs. Frequent testing and planning ahead will cause fixing the program to be easy from the beginning.

**Update Risk**

The plan is to decide whether we want to update the program or create a new one and how we will go on doing that in order to make the final product ready in time.

**Tech Risk**

The plan is to research for similar programs and available tools that can help us manufacture this program to the best of our ability.

**COVID Risk**

The plan is to take safety precautions, wear masks in public, keep social distance, and limit in-person interactions to stay safe and not contract COVID-19.

**Online Communication Risk**

The plan is to work through online tools such as Discord, Zoom, and Google Drive to develop the program together virtually since we have very limited in-person meetings.