



Sprint 1: Risk Management Log

By:

Camil Bouzidi (40099611)

Radley Carpio (40074888)

Bicher Chammaa (40096200)

Nimit Jaggi (40032159)

Matthew Kevork (40063824)

Cedric Martens (40086877)

William Morin-Laberge (40097269)

Adrien Tremblay (40108982)

Pierre-Olivier Trottier (40059235)

A report submitted in partial fulfillment of SOEN390.

Concordia University

February 3rd, 2021

Introduction

In this report, an initial risk analysis for the EPIC Resource Planner software product is described and explored. Given that this is the first development sprint related to the project, many of the risks outlined in this document are subject to change. This preliminary exploration of risk analysis has the goal of setting a baseline for upcoming sprints and meetings with stakeholders. First, different categories of risks will be explored. Then, for each of these categories, risks will be exposed through different elicitation techniques. Each of these risks will then be assigned qualitative values, which will then be turned into quantitative values to be used to rank these risks and see which will be tackled first. In order to tackle the risk, one of the techniques outlined in the risk management plan will be applied.

The following information is to be maintained for every risk (all of which is explained in the Risk Management Plan):

- Description
- Date of identification
- Probability
- Impact
- Rating/Priority
- Response Strategy
- Status
- Owner (to be defined in later sprints)

Identification

Risk Categories

After deliberation and discussion with the product owner, the stakeholders, the development team as well as an exploration of the problem domain, the following risk categories have been chosen:

- **Administrative or Process:** This has to do with staff management, meetings, stakeholders and any other situation where an error in the code or the problem domain is not to blame.
- **Code or QA:** This has to do with any framework, API or coding defect or issue. This includes the bugs that arise as a result of testing.
- **Problem Domain:** While it is not the software development team's responsibility to run the business that uses the EPIC Resource Planner, the team should still analyze the problem domain and highlight any risks born out of the software's interaction with the problem domain.

Risks

For the previously mentioned categories, the following risks have been found:

Administrative/Process Risks

- **Risk 1.1:** Developer drops out of the team.
 - **Description:** for any reason, a developer drops out of the team and does not work on the project anymore.
 - **Date of identification:** January 17th, 2021
 - **Probability:** High (50%)
 - **Impact:** Medium (0.35)
 - **Rating:** 0.175

- Status: Potential
 - Response Strategy:
 - Mitigation: Developers shall perform as much knowledge sharing that they can to avoid losing knowledge when a developer leaves the team.
 - Transfer: The SOEN 390 administration will have to change the scope of the project if necessary.
-
- **Risk 1.2:** Developer must be relieved of work because of illness or other circumstances.
 - Description: Should a developer need some time away from the project due to an illness or circumstances outside of their control, their share of work must be handled.
 - Date of identification: January 17th, 2021
 - Probability: Very High (80%)
 - Impact: Medium (0.35)
 - Rating: 0.28
 - Status: Potential
 - Response Strategies:
 - Mitigation: Developers should have high visibility on others' work through weekly meetings. Should very complicated features need to be implemented, pair programming will be implemented.
 - Transfer: The SOEN 390 administration will have to change the requirements of the project if necessary.
-
- **Risk 1.3:** Feature will not be delivered on time.
 - Description: If a feature is planned for a sprint and cannot be delivered by the end of the sprint, it is late.
 - Date of Identification: January 17th, 2021
 - Probability: High (50%)
 - Impact: High (0.5)
 - Rating: 0.25
 - Status: Potential
 - Response Strategies:

- Mitigation: Features will be ranked by size to get a better idea of how they should be prioritized. Once prioritized, if a feature is not delivered at the end of the sprint, it will be implemented by the end of the next sprint.
- **Risk 1.4: Poor Estimations**
 - Description: The estimation of size and effort needed for various features and user stories are off, leading some features to being pushed to other sprints.
 - Date of Identification: January 28th, 2021
 - Probability: High (50%)
 - Impact: High (0.5)
 - Rating: 0.250
 - Status: Potential
 - Response Strategies:
 - Mitigation: As the project goes on and the development team becomes more comfortable with the technology and development tools, the estimations made by the development team will be more accurate both in terms of size and length. In order to accelerate this evolution, any task whose estimation was wrong will be reviewed by the development team at the end of the sprint in order to understand why or how it was misestimated.
- **Risk 1.5: Poor risk management**
 - Description: The initial plan for the risk management plan is unclear or leads to a poor risk management strategy where some risks go completely untouched and slow the production of the team.
 - Date of identification: January 20th, 2021
 - Probability: Medium (35%)
 - Impact: Medium (0.35)
 - Rating: 0.1225
 - Status: Potential
 - Response Strategies:

- Mitigation: The risk management plan is to be revisited as soon as a problem is noticed by the development team. A meeting will be called in order to develop new metrics or methodologies related to risk management.
- **Risk 1.6:** Sudden growth in requirements
 - Description: It is possible that stakeholders ask for the development team to suddenly develop new features and have additional requirements.
 - Date of identification: January 20th, 2021
 - Probability: Low (20%)
 - Impact: Very High (0.8)
 - Rating: 0.160
 - Status: Possible
 - Response Strategies:
 - Mitigation: Meetings with product owners are scheduled for every sprint. The development team will also constantly refer to the requirements document to verify whether or not a new requirement falls within the guidelines previously outlined.
 - Accept: Should requirements not fall within the previously agreed upon guidelines, the software development team will tell product owners or stakeholders that the feature will not be implemented.

Code or QA

- **Risk 2.1:** Poor End-user Engagement
 - Description: The product does not incorporate enough user feedback into its design. As a result, the end-user is unlikely to be satisfied with the product and is unlikely to adopt it.
 - Date of Identification: January 17th, 2021
 - Probability: Medium (35%)
 - Impact: Very High (0.75)
 - Rating: 0.2625
 - Status: Monitoring
 - Response Strategies:

- Mitigation: More user testing, surveys, focus groups should take place to gain a better understanding of the user's requirements and expectations. Users should also be given working prototypes for their feedback.
- **Risk 2.2: Poor Quality code**
 - Description: The codebase is of poor quality and puts the project into technical debt reducing the agility of the project. Code may be of poor quality because it is hard to read, inefficient, lacks good tests, etc... Poor quality code can lead the codebase to becoming needlessly hard to maintain, expand upon and a product that is of poor performance or bug prone.
 - Date of Identification: January 17th, 2021
 - Probability: Medium (25%)
 - Impact: Medium (0.35)
 - Rating: 0.0375
 - Status: Potential
 - Response Strategies:
 - Mitigation: More code reviews, software testing, static verification should be done. More software testers could potentially be onboarded for the project. Additionally, a clear coding standard could be put in place for all developers to follow.
 - Transference: Software QA could potentially be outsourced to another development team.
- **Risk 2.3: Compromising on Designs**
 - Description: The developers rush software design and architecture to jump straight into coding. This can result in a less agile, extensible, and performant end-product. This often leads to a lot of work going to waste because of constantly changing software architecture or design.
 - Date of Identification: January 17th, 2021
 - Probability: Medium (30%)
 - Impact: Medium (0.35)
 - Rating: 0.105

- Status: Potential
- Response Strategies:
 - Mitigation: Software architecture and design experts could be onboarded. The development team should seek approval for architecture and design from tech leads or management. An Agile approach to software design could be implemented where software is designed iteratively.
 - Transference: Software Architecture could potentially be outsourced to another development team.
- **Risk 2.4: Gold Plating**
 - Description: Gold plating is when developers waste time implementing features not specified in the functional requirements or waste time writing code based on a personal unapproved codebase decision they have taken.
 - Date of Identification: January 17th, 2021
 - Probability: High (60%)
 - Impact: Low (0.2)
 - Rating: 0.12
 - Status: Potential
 - Response Strategies:
 - Mitigation: Developers who do Gold Plating should be severely punished to deter future instances of this occurring.
 - Mitigation: To provide developers with a clear understanding on which features are to be implemented and how, each feature should be made into a user story, and each user story should be broken into substories.
 - Acceptance: Developers will always make decision while coding. There is no clear line between what decision is acceptable to be taken by a developer on the fly, and a decision that needs to be approved by other developers. A few instances of gold-plating will always be unavoidable and is acceptable.

Problem Domain

- **Risk 3.1: Market shifts**

- Description: The market shifts resulting in a change in requirements or product viability. For EPIC Resource Planner this could mean that bicycles become less popular, or certain types of bicycles shift in popularity.
 - Date of Identification: January 17th, 2021
 - Probability: Low (5%)
 - Impact: Medium (0.35)
 - Rating: 0.0175
 - Status: Potential
 - Response Strategies:
 - Acceptance: When the change to the market is relatively small not much can be done, and the risk should simply be accepted.
- **Risk 3.2: Poor domain understanding**
 - Description: The software team does not understand the problem domain well enough, and this is reflected by incomplete or inaccurate software requirements.
 - Date of Identification: January 17th, 2021
 - Probability: Medium (40%)
 - Impact: High (0.5)
 - Rating: 0.2
 - Status: Potential
 - Response Strategies:
 - Mitigation: More market research should be done to build a better understanding of the domain problem. Additionally, domain experts could be hired.
- **Risk 3.3: Legal risk**
 - Description: The project needs to abide by laws or regulations. If these laws arise during the project or are not identified an early stage of the project development, there could be a substation deterrent to the product in terms of needing to reevaluate requirements.
 - Date of Identification: January 17th, 2021
 - Probability: Low (15%)

- Impact: Low (0.2)
 - Rating: 0.03
 - Status: Potential
 - Response Strategies:
 - Mitigation: The law should be thoroughly examined to find any laws that may have bearing on the project.
- **Risk 3.4: External hazards**
 - Description: External hazards are catastrophic events such as floods or earthquakes that may severely impact project assets or staff.
 - Date of Identification: January 17th, 2021
 - Probability: Low (1%)
 - Impact: Very High (0.8)
 - Rating: 0.008
 - Status: Potential
 - Response Strategies:
 - Mitigation: There should be stringent safety measures in place in the event of any catastrophic event to minimize impact caused by such an event.

Qualitative Risk Analysis Table

| Impact | Very High | 1.6, 3.4 | 2.1 | | |
|-------------|-----------|----------|---------------|----------|-----------|
| | High | 2.4 | 3.2 | 1.3, 1.4 | |
| | Medium | 3.1 | 1.5, 2.2, 2.3 | 1.1, 1.2 | |
| | Low | 3.3 | | | |
| | | Low | Medium | High | Very High |
| Probability | | | | | |