BoomBook Bookstore
Risk Management Report

1   Introduction

This report includes detailed information about risks and risk management. The risks can occur during the development of the project, even after deployment. The instant actions or mitigation against to happening of risks are determined to prevent the risks transforming to a problem.

The magnitude of risks is numerated as possibility x impact number. The impact number is numerated like this:

- **Catastrophic**: 4

- **Serious**:   3

- **Tolerable**:   2

- **Insignificant**: 1

Magnitude calculated with **possibility** and **impact** number. The maximum risk can be 1.0 x 4 = 4 which is not forecasted in the project.

The forecasted risks and observed lists with before its happening possibilities are listed in "**Description"** part.

The risks are listed with options and activities according to iteration model in **"Risk Management Report Specifications"** part.

2   Description

2.1 Redundant dependency

- o **Magnitude**: 2.0

2.2 Conflict version on repository

- o **Magnitude:** 1.8

2.3 Incompatible tool with user's environment

- o **Magnitude:** 1.5

2.4 Underestimated evaluation

- o **Magnitude:** 1.4

2.5 Insufficient performance

- o **Magnitude:** 1.2

2.6 Underestimated time

- o **Magnitude:** 1.2

2.7 Changing requirements

- o **Magnitude:** 1.2

2.8 Non-optimal task dependencies

- o **Magnitude:** 1.0

2.9 Unavailable staff

- o **Magnitude:** 0.8

2.10 Incompatible software framework

- o **Magnitude:** 0.8

2.11 Low transaction speed per second

- o **Magnitude:** 0.8

2.12 Incompatible database design

- o **Magnitude:** 0.6

2.13 Crashed repository

- o **Magnitude:** 0.5

2.14 Vulnerabilities of system

- o **Magnitude:** 0.5

2.15 Insufficient staff

- o **Magnitude:** 0.4

2.16 Incompatible development environment

- o **Magnitude:** 0.4

2.17 Unavailable physical meeting environment

- o **Magnitude:** 0.1

3   Risk Management Report Specifications

3.1 Risk Not Forecasted but Observed in Planning

Risk ID	Date Identified	Headline	Description	Type	Impact	Probability	Magnitude	Owner	Mitigation Strategy
1	02.05.2020	Incompatible database design	Incompatible the implementation of database system or a table.	Technology	3	20%	0.6	Database Architect - Software Developer	The database architecture communicated with developer so that the effects on web development part were handled immediately.
2	02.05.2020	Unavailable physical meeting environment	A chaotic situations are happened during development process curfew because of the pandemi.	Organizational	2	5%	0.1	Software Project Manager	Our meetings are evaluated on Discord application which is an online environment.
3	02.05.2020	Insufficient staff	A staff was unknowing will be used technology.	People	2	20%	0.4	Software Project Manager	The manager determined an available staff to train insufficient staff which can be possible to learn in consistent time. Other

3.2 Risk Forecasted in Planning

Risk ID	Date Identified	Headline	Description	Type	Impact	Probability	Magnitude	Owner	Mitigation Strategy
4	02.05.2020	Unavailable staff	Staff is unavailable while other member has dependence task on project.	People	2	40%	0.8	Software Project Manager – Software Developer	The manager check for an available staff for this task. If there is no available person, the task is apportioned to all developer
5	02.05.2020	Underestimated time	Required time for a deliverable of the project is not estimated correctly.	Estimation	4	30%	1.2	Software Project Manager	The manager must immediately change the initial plan according to deadline and increase the effort increase work time per a day.
6	02.05.2020	Incompatible software framework	The software framework which is decided to use is out of date or incompatible with architect, environment or other frameworks on system.	Technology	4	20%	0.8	Software Architect	The architect immediately changes the technology with proper and not affect on changes other components that were integrated already.
7	02.05.2020	Insufficient performance	The response time to user could be slower than expected after deployment to usage.	Technology	3	40%	1.2	Software Analyst – Software Developer	The analyst should found cause of the low performance and report to interested person who is responsible from the problematic part of the software.
8	02.05.2020	Crashed repository	Repository to used to compose all software project can be crashed and lost all versions of project.	Technology	5	10%	0.5	Software Configuration Manager	The configuration manager should back-up all version in different storage environment from version control system according to this risk.

9	02.05.2020	Low transaction speed per second	Transactions can be conflicted or not applied the results to server on time.	Technology	2	40%	0.8	Software Developer - Database Architect	The software architect who is responsible the low transaction speed module of the software, should check transactions, if determines an error from server side, should report to the database architect.
10	02.05.2020	Changing requirements	Some of the stakeholders can demand changing functional requirements from the user side or system side.	Requirements	2	60%	1.2	Software Analyst	The software analyst report to developers about the necessary changing implemantation of changed requirements to integrate.
11	02.05.2020	Incompatible development environment	Development environment such as IDE or operating system is unsufficient for development of the software.	Tools	2	20%	0.4	Software Analyst	The analyst must find the most compatible and user-friendly environment to prevent hardness the learned new environment to develop the project.
12	02.05.2020	Conflict version on repository	Developer can cause commit old version component or unversioned component such as incompatible code/component with other components	Technology	3	60%	1.8	Software Configuration Manager - Software Developer	At least one staff must view the commits to preveten this, or if too late, then configuration manager restore backup from used version control system, check last version commits and recompose these components each other.

13	02.05.2020	Underestimated evaluation	Appear some underestimated evaluation after evaluation part according to feeadbacks from user side.	Technology	2	70%	1.4	Sofware Tester	The tester must analyze all possible bugs and determine and apply unit tests repeatedly and give reports to developers during the development process.
14	02.05.2020	Redundant Dependency	The project can depend too many depencies that are unnecessary. Changing in these dependencies causes errors and to need continously maintanance.	Technology	3	65%	2.0	Software Analyst	The analyst should be aware of what kind of dependencies are needed to use. She/he must eliminate the redundants before the implemantation process.
15	02.05.2020	Non-optimal task dependencies	The dependencies between tasks can be defined insufficently. It restricts to parallel processing of tasks.	Organizational	2	50%	1.0	Software Project Manager	The manager must be aware of inputs and outputs of tasks in between them.
16	02.05.2020	Incompatible tool with user's environment	The used tools in integrated aprt of project are not compatible user's environment. In our case, a web browser can be incompatible with some front-end technologies.	Tools	3	50%	1.5	Software Tester	The related unit test must be determined and tested by the tester in the development process.
17	02.05.2020	Vulnerabilities of system	The security vulnerabilities against attackers can exist on the system to capture user information or to just prevent working properly.	Technology	2	25%	0.5	Software analyst	Many sufficient dependencies are available to integrate to existent software. The analyzer can find one of them in design part.