**The Green Git Inspector**

**Risk Register**

**FIT2100 Semester 2, 2019**

**Min Hee Cho, Jack Thomas Whelan, Jun Yoong Ooi, Abhijeet Mondal**

**TEAM green**

**This document contains a register of risks relating to the project. Each risk is accompanied by an estimated impact and likelihood, a strategy for monitoring the state of the risk, and a mitigation plan for that risk. The estimated impact and likelihood is measured using a 5 point system (1 being the lowest, 5 being the highest) as per the course notes. Succeeding the register of risks, is a risk matrix displaying roughly how pressing each risk is.**

**Risk Register:**

**Risk 1: Unavailability of the client during important time hinders progress**

**Description:** risk that progress will be delayed or halted due to inability to consult with client due to their unavailability.

**High Impact(4), Nominal probability(2)**

**Monitoring strategy:** Check the availability of the client 3 days prior to scheduled meetings.

**Mitigation plan:** Establish that the product owner will make decisions on behalf of the client if the client is unavailable during scheduled meetings.

**Risk 2: Github changes API, resulting in incompatibilities between the software and Github**

**Description:** risk that the software being developed may be non-functional if the Github REST API changes during the project.

**High Impact(4), Low Probability(1)**

**Monitoring strategy:**

Monitor the Github REST API website weekly or daily during sprints to keep track of any potential changes in their REST API.

**Mitigation plan:**

Implement a design for the software which has a layer of abstraction between the main functionality and the part that utilises the Github REST API. This will mean that less work would be required to update the software if the Github API were to change.

**Risk 3: Github crashes/malfunctions hindering progress with the project**

**Description:** risk that progress will be delayed due to both inability to access the project repository and use git, and the inability to test/run the part of the software that interacts with Github.

**Extreme Impact(5), Low probability(1)**

**Monitoring strategy:**

Monitor the status of the Github’s performance weekly, or daily during sprints. This can be done easily by visiting <https://www.githubstatus.com/> and checking that all necessary functionality of Github is operational.

**Mitigation plan:**

**- Organizational**

Alternative methods of sharing source code and documents between team members will be ready to use if necessary. These include but are not limited to: Mercurial, a shared google drive, emailing.

**- Technical**

Implement a design for the software which has a layer of abstraction between the main functionality and the part that utilises the Github REST API. This means that if Github was not operational at a crucial time in development, the rest of the software could still be developed, tested, and presented to the client.

**Risk 4: Requirements changing**

**Description:** risk that time/effort will be wasted in development of features that are no longer required by the client.

**Moderate Impact(3), Nominal Probability(2)**

**Monitoring strategy:**

Current status of the software along with current PBI’s will be presented to and discussed with the client during meetings. This will keep the team informed of any changes to the requirements.

**Mitigation plan:**

The process model that is to be employed by the team caters well for any changes in requirements. All features that have been implemented and those that are to be implemented later on will be discussed with the client during meetings. This gives the client multiple opportunities to change the requirements without inconveniencing the team or wasting much time/effort in developing.

**Risk 5: Unfeasible requirements results in wasted time and incompleteness of the project**

**Description:** risk that time/effort will be wasted in attempting to implement features that are infeasible for unforeseen circumstances.

**High Impact(4), Low Probability(1)**

**Monitoring strategy:**

Develop spikes, or test code for new features/functionality before fully implementing them. This will make it clear whether a feature is feasible or not.

**Mitigation plan:**

Develop spikes, or test code for new features/functionality before fully implementing it. This way, if a feature is unfeasible, less time will be wasted and the team can continue working on other parts of the requirements.

**Risk 6: Unavailability of the team hinders progress**

**Description:** risk that progress is minimal due to the unavailability of one or more team members.

**Nominal Impact(2), Nominal Probability(2)**

**Monitoring strategy:**

Check that all team members are available for meeting 3 days prior to the scheduled meeting.

**Mitigation plan:**

Arrange alternative times for meetings based on the schedules of team members (and client if necessary). Meetings will be rescheduled to the alternative times if team members are unavailable for the regular meetings.

**Risk 7: Poor productivity leads to incompletion of scheduled tasks**

**Description:** risk that features are not implemented due to poor productivity and time management of the team.

**High Impact(4), Nominal Probability(2)**

**Monitoring strategy:**

A burndown chart will be used to monitor the team’s progress. The burndown chart will show the rate at which the team is completing user stories during sprints. It will map the number of story points completed by the team against the number of days into the sprint. This will make it clear how productive the team is being compared to how productive the team needs to be to complete all items in the sprint backlog.

**Mitigation plan:**

Using the burndown chart, it will be more clear to the team members how fast they must be completing the user stories in the sprint backlog. Along with this, some basic strategies for better productivity within the team will be used. This includes, delegating tasks to team members who are more experienced in that area, changing task allocation if a task proves to be too difficult for a certain team member, and regular meetings with team members to discuss approaches to certain tasks.

**Risk 8: Incapability of developers hinders progress with the project**

**Description:** risk that progress is minimal due to the incapabilities of the team members to implement required features.

**High Impact(4), Nominal Probability(2)**

**Monitoring strategy:**

Daily meetings will be conducted during sprints. During which, any team members that are struggling have the opportunity to raise the issue with the team.

**Mitigation plan:**

A number of different strategies will be put into place to reduce time wasted due to incapabilities of the team. These include: delegating tasks to team members who are more experienced in that area, changing task allocation if a task proves to be too difficult for a certain team member, and regular meetings with team members to discuss approaches to certain tasks.

**Risk Matrix:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Impact** | **5** | **3** |  |  |  |  |
| **4** | **2, 5** | **1, 7, 8** |  |  |  |
| **3** |  | **4** |  |  |  |
| **2** |  | **6** |  |  |  |
| **1** |  |  |  |  |  |
|  | **1** | **2** | **3** | **4** | **5** |
|  | **Probability** | | | | | |

**Legend:**

|  |  |
| --- | --- |
| **Urgency of risk** | |
|  | **Low** |
|  | **Medium** |
|  | **High** |
|  | **Extreme** |