CSCI311 - Draft Assignment

Executive Summary

The Python community is a highly skilled, incredibly talented one that endeavours to enhance, improve and further the use of the Python language. As part of this community effort, the Python community utilises a variety of tools to co-ordinate and test bug fixes as well as engage in discussions through mailing lists and internet forums.

This project aims to liberate data that is currently 'locked away' in the above data sources by developing an application to visualise existing and new data, correlate events such as bug reports with the most appropriate answer and allow interaction between all members involved in the community.

Risks and Mitigation Strategies

There are several risks that may effect this project and the deliverables are developed. These risks can be neutralised (if possible) or strategically managed to minimise their impact. Risks include:

1. Risks

1.1. <u>Over-promise</u>

1.2. Risk Magnitude or Ranking

High

1.3. Description

Developers may promise a feature that is technically difficult to implement or maintain.

1.4. Impacts

"Too much time" (as defined in the Main Report "Abbreviations and Definitions" section) spent on the implementing a feature may result in the team being unable to create functions that depend on the problematic feature. It may also result in a delay in delivery of project if such feature-requests are unchecked before being promised.

1.5. Indicators

Failure to meet internal milestones for various functions being created whilst the team (or single team member) is focused on solving a problem. A simple threshold would be "too much time" (as defined in the Main Report "Abbreviations and Definitions" section) between initial and final commit.

1.6. Mitigation Strategy

Review architecture of the program and calculate if a problem can be broken down into smaller sub-problems that are easier to solve within smaller time frames.

1.7. Contingency Plan

Cease all work on the current function and have the team re-evaluate the current feature. If it is critical or required for other functions to work, then the approach to implementation may need to be reviewed. If it is not necessary, then a removal of functionality or feature may be needed.`

1.8. Under-deliver

1.8.1. Risk Magnitude or Ranking

High

1.8.2. Description

Developers may fail to implement a feature that was specified in the system requirements report.

1.8.3. Impacts

If the system delivered fails to meet the system requirements report, then the client may:

- Cancel the project - pay what was created and spread bad word about the business
- Ask for it to be implemented at no extra cost - request the system be kept but that it be implemented without payment as it was the original specification

The impact on the developer team may be overtime spent to update the system to meet the requirements as well as the team running at a loss (of finances) due to incompetence or inability.

1.8.4. Indicators

Failure to accurately the progress of all team members by not checking commit comments, or bringing up a single member's "weighing down of the chain" approach to assisting the group.

1.8.5. Mitigation Strategy

Ensure a weekly plan is implemented with achievable goals and communicate with the client each week with a status update. This will ensure maximum communication and that the team is operating within acceptable time-constraints.

1.8.6. Contingency Plan

Do as the client asks, unless their request is technically unreasonable or is beyond the scope of abilities that the team possess.

1.9. Lack of Attendance

1.9.1. Risk Magnitude or Ranking

Medium

1.9.2. Description

If a developer or client fails to attend meetings, a lack of highly-important information and time for clarification may be missed.

1.9.3. Impacts

If a developer is consistently missing meetings for no acceptable reason, there may be reason to eject them from the group if their work is impacted.

If a client misses too many meetings, the developers may begin to create a product that is not in-keeping with the client's understanding of how the product works. This may lead to Risk 2.2 being realised.

Finally, a major impact is that developers (and to an extent, clientele) who are away must be kept up-to-date with minutes of the meeting, resulting in a loss productivity as they must re-examine the contents of the meeting as well as clarify the minutes with an attending person to make sure they have a sound understanding of what was discussed.

1.9.4. Indicators

Reviewing meeting minutes and tracking 'apologies' or 'missing attendees' electronically.
Generating a report or chart to see who is missing meetings constantly.

1.9.5. Mitigation Strategy

Ensure a suitable meeting time is chosen well-ahead of the actual meeting time or allow remote conferencing if required.

1.9.6. Contingency Plan

Discuss a better time or initiate a way to minimise the time required to review meeting minutes.

1.10. Failure To Deliver By Deadline

1.10.1. Risk Magnitude or Ranking

Critical

1.10.2. Description

The developer team may fail to deliver the project by the specified deadline.

1.10.3. Impacts

The client may request the project be cancelled, demand a refund, or request the project be completed without payment (citing the original specification).
The developers may have to complete the project with no payment and/or with overtime to meet a new deadline.

1.10.4. Indicators

Monitor indicators mentioned in 2.1.4 and 2.2.4.

1.10.5. Mitigation Strategy

Similar to items in 2.1.5 and 2.2.5, an addition to these mitigation strategies would be to ensure that all team members have their well-being checked (that is, ensure they are not experiencing any personal difficulties that may impact their ability to complete assigned work) as well as constant team meetings to discuss any issues that may cause a hold-up in the delivery of the project.

1.10.6. Contingency Plan

Do as the client asks, unless their request is technically unreasonable or is beyond the scope of abilities that the team possess.

1.11. Miscommunications and Misunderstandings

1.11.1. Risk Magnitude or Ranking

High

1.11.2. Description

The developer team may implement a process in a way that does not comply with the client's understanding.

1.11.3. Impacts

The development team may begin implementing a feature incorrectly, resulting in a loss of time due to a misunderstanding.

Miscommunication may result in breakdown of weekly updates between team and client or differing views of assigned tasks amongst developers.

1.11.4. Indicators

Failure to attend meetings (as per 2.3) is one indicator to check for. Constant rebuttals from the client in how the application should work may indicate a communication issue. Picking up on constant, minute changes may require a review of the system (or feature) being implemented.

1.11.5. Mitigation Strategy

Ensure the client is specific in their requirements, that development team communicates their understanding and interpretation of what the client wants and that the client signs off after each clarification for documentation purposes.

1.11.6. Contingency Plan

Organise an emergency meeting with the client to minimise misunderstandings and ensure the development team can make the necessary changes in a timely manner. Communicate with the client any changes and ensure that a proper framework (such as weekly or even daily updates) is created and followed.