Incremental Deliverable 3 Risk Report

Daniel Benesh - CMPT 371 - Team 3

1. Deliverable Deadlines and Incomplete Work: Probability: 7; Impact: 9; Recurring

   Throughout our development process we have become increasingly ambitious in the amount of progress we attempt to make. The previous deliverables have been mainly focused on UI implementation, Linting, CI pipeline and setting ourselves up for success. In the current deliverable we have attempted to implement the more functional elements of our project. This has come with some difficulties. The way this risk materialized is in affecting the timeline for this deliverable. Our contingency plan has been to push the tasks to the following deliverable. To help mitigate the risk in the future, we have done planning poker to justify estimations for each task. In the areas that we were successful in completing tasks, we have found pair programming to be invaluable. In practicing pair programming, we have found that some benefit was left to be gained. Pair programming will be an essential practice going forward. There are also communication issues that have played a role in the incomplete work. We plan to mitigate the risk of this happening again by strengthening these practices.

2. Communication: Probability:8; Impact 10; Recurring

   We are still working on more productive ways to communicate. Over the midterm break, some of our communication has broken down. The daily standup bot hadn't been used as much as it should have been. The bot has both positive and negative effects. The positive is that it can keep us accountable, but it must be used for us to benefit from it. We have found that it to be useful but it should not be used as a replacement for actually communicating directly between members. Leads need to communicate with their team members more effectively. They should know the progress status of their team members. Communication also goes both ways. Leads should also communicate between one another to ensure that there are no surprises at code freeze. The status of all features should be communicated between group members. If a lead does not hear from members working on the team in a reasonable amount of time, they should inquire on the status of their work. The fallout from not communicating effectively results in incomplete work and changes the timeline of the project. This affects the progress of the project. With two deliverables to go, it is essential that we communicate more effectively.

3. New technologies: Probability: 9, Impact: 7; Recurring

   Many of our team members are still unfamiliar with the technologies that we are implementing in this project. These include Linting, React, Typescript, CSS, etc. This leads to low productivity in completing our tasks. With low productivity it affects the quality of code, time to complete and other factors that effectively halt production. To mitigate this risk we have been implementing pair programming. Members are able to help one another solve problems and share knowledge from more experienced programmers in these new technologies. Members need to communicate their issues as well. Tutorials have also been posted in Notion to serve as guides to help us complete our tasks. We are all still learning how to use these new technologies and hope that with these measures we are able to complete our work in a reasonable time frame.

4. Time estimations: Probability: 5, Impact: 4; Recurring

   Time estimations have been hard to gauge for us in this deliverable. We are trying to mitigate this risk, which results in incomplete work, by planning poker. This past week we had all contributed to the exercise. During this activity we went through all Github issues and made a case for the amount of time we believe the task would take to complete. . In planning poker, we hope to improve our estimations on implementing features of our App. If there are hindrances to completing tasks, leads should be aware of these deficiencies. As mentioned above we have been struggling with communication. The leads need to be engaged with team members to ensure they are aware of the progress being made and what struggles they are running into while attempting to complete their task. We include this issue under the time estimation risk because they go hand in hand. We need to have more proactivity between team members to ensure that tasks are completed. When the risk of poor time estimations materialize, work is not completed on the time set out, and it results in incomplete work. This impedes the progress on our project.

5. Backend Codebase: Probability: 8, Impact: 8; Recurring

   We are beginning to find more and more issues while working with the backend codebase. We have found that the ‘delete’ API call, to remove data submitted to the database is non-existent and we will need to code it ourselves. This makes removing data from the database not possible. When starting the page locally while we are working on it, there is a time limit on how long the page will function resulting in the “white screen of death”. The reason is that we cannot call an event handler while rendering the page. The workaround has been to delete the local storage in the web page itself through browser dev tools. The libraries being used for sign in are userLoggedIn and useReadLocalStorage. Also we have found a bug where the system does not deal with data from AppleWatch, although it does work with Fitbit. We will need to fix these bugs going forward. The result of this issue is that development is slowed because of a non-functional page. To mitigate these risks our team has been familiarizing ourselves with the codebase and searching for deficiencies that will affect the productivity of our developers.

6. Risks of Being Unaware of Risks: Probability:6; Impact: 8

   If people are unaware of the risks involved with this project it can result in poor productivity impacting our timeline. If someone is unaware that they need to communicate more, then how can they improve? If they are unfamiliar with the backend API calls and the associated bugs, they will have trouble finding the bug. If members are unaware that some features have not been implemented, how can they test them? The risk of being unaware of risks, can lead to a complete halt in productivity. If the right hand doesn't know what the left is doing, it will lead to confusion. We intend to have risk meetings during the next deliverable so that everyone is clear on the problems associated with this project. In doing this we intend to spread knowledge and make the team more productive.

7. QA Productivity during the start of Deliverable: Probability: 7; Impact: 5

   We have found that we need to keep the QA team able to begin testing earlier. During the past deliverables, we have not had PR's early enough for them to begin testing. Instead, they are left to begin testing late into the deliverable after code freeze. This results in poor productivity on the QA side because we are leaving them idle while we could be giving them code to test. If they get the code earlier, they will be able to provide better testing. They will be able to work with the developer directly and fix issues quicker. To mitigate this risk, we will be making more frequent PR's to get them code to look at before code freeze in hopes to become more productive.

8. Handling large amounts of data: Probability 8; Impact: 7

   We are still not ready to implement a spike prototype to handle large amounts of data due to some of the features not being implemented. We need to have the site in a working state before attempting to change the architecture of our API calls to the database. This is an important implementation to deal with big data. The risk involved in getting it to work is reorganizing how our system will access, store and update data with a new database. This will involve using new API calls to work with the fitness data. This new framework will reorganize the structure of our application significantly while improving the way in which it handles data. We will not roll out this framework to our project until we are able to confirm that it is working. To mitigate the risk, knowledge sharing through tutorials will be implemented to ensure that the framework is well understood for those developing the spike prototype. We will also have to do more research to materialize this overhaul.

9. Members Becoming Ill: Probability: 7, Impact:7

   When a group member gets ill they will have a limited capacity to finish tasks. This will result in low productivity and tasks not being accomplished. The plan to mitigate this risk is to implement shadowing other team members.If something were to happen to team leads, it will impact the entirety of how the group functions. Members will need to step into those roles to ensure that the work we set out to accomplish is accomplished. For the project lead, getting updates from Dev and QA leads, making sure the activity board of members are filled out, meeting times with the stakeholder and professor, among other things. For the QA lead, making sure to get updates from team members, updating the testing matrix, and other testing concerns. For the Dev lead, getting updates from team members, helping work out coding issues. In shadowing these key members we will be able to mitigate the issues arising from an absent team member in a key role.

10. Documentation: 5, Impact: 6

    As mentioned in the Backend risks we will need to understand the codebase more thouroughly. In this deliverable, we intend to make the backend more understandable to all group members. To do this we will need to start documenting the code that we haven't written. The risk of us not having a complete understanding of the codebase will result in low productivity. We need to be actively searching for deficiencies in the codebase. We will need to add clarification to it with comments and using JSDoc's. This will add clarity to the codebase making it easier to work with.