Risk Mitigation and Planning - A7

1. (10 pts) Personnel shortfalls

Risk Likelihood – 10

Risk Impact (without mitigation) – 10

Risk Impact (with mitigation) - 3

Our project requires a large amount of developers in order to re-vamp the Kayak website. Because of our high dependence on personnel it is almost certain that we will run into issues such as a difference in experience and skills, appropriateness of experience, staff satisfaction, and staff turnover rates.

In order to mitigate these issues we must insure that our hiring process includes technical questions in order to acquire a base level of developer and that communication inside the operation is encouraged so those with higher skill levels will be able to help the others learn. Staff satisfaction is something that many tech companies accomplish by having a laid back, comfortable work environment so your employees never feel like leaving even though you are only paying them to be there for forty hours a week. This keeps them happy and gets you even more bang for your buck.

Staff turnover problems can be downsized by ensuring constant influx of new employees, as with all tech companies turnover is guaranteed at an almost extremely predictable rate. Also ensuring code review and commenting practices will allow for a spread out knowledge of the entire codebase so when one important employee leaves others are still able to do the work.

2. (10 pts) Unrealistic schedules and budgets

Risk Likelihood – 8

Risk Impact (without mitigation) – 8

Risk Impact (with mitigation) - 3

Unrealistic schedules and budgets are very dangerous for external companies bidding for software projects. Because we had to bid for this project for Kayak, it can be assumed we tried to minimize the cost and timeline to where it is right at the border of our abilities. Without mitigation our project could easily get over budget and late on schedule.

In order to mitigate these issues we are lucky that we have a pdm diagram from assignment five that we can follow exactly to stay on schedule. It will be important that we do not fall behind the latest start / latest finish, and can use the early start early finish in order to schedule our developer time so even if they fall behind schedule, they aren’t actually behind schedule.

3. (10 pts) Developing the wrong software functions

Risk Likelihood – 3

Risk Impact (without mitigation) – 5

Risk Impact (with mitigation) - 1

This issue is very common when there is not a fully requirements documentation for a project. If you don’t know what the customer wants it is very easy to do something that you think is correct, and might actually be correct, but when it comes time to get your feedback you are shocked with what response you get because the customer isn’t happy at all!

This can be easily mitigated by a thorough requirements process that we can follow that is legally binding for both parties. This way the customer needs to say what they want and they know that what they request is exactly what they are going to get. Specifically for us, because we are doing mostly a visual revamp we will not touch very much software functionality so we are pretty safe here.

4. (10 pts) Developing the wrong user interface

Risk Likelihood – 5

Risk Impact (without mitigation) – 7

Risk Impact (with mitigation) - 2

This issue is very similar to developing the wrong software functions, and we can assume what we have said about that carries over, except for the fact that we are doing a visual revamp and developing the wrong user interfaces is a bigger possibility.

5. (10 pts) Gold plating

Risk Likelihood – 5

Risk Impact (without mitigation) – 3

Risk Impact (with mitigation) - 1

Gold plating is when you develop things that are out of scope that might be beneficial in the future or are just purely extravagant. For our project we could end up making their website look great, but waste time on making things work beyond the requirements so that developers don’t have time to finish other requirements that are left.

In order to lower this risk it’s important that developers know how long they should be spending on each task and that when tasks are given out we use the requirements to write out details expectations of what we want exactly from the developers. On some teams there are members entirely dedicated to splitting up the requirements and getting those tasks planned out, ready, and estimated for developers in order so they know exactly what they’re working on and if we were to implements this strategy we would be much safer.

6. (10 pts) Continuing stream of requirements changes

Risk Likelihood – 4

Risk Impact (without mitigation) – 4

Risk Impact (with mitigation) – 2

When requirements change constantly it makes it extremely hard to plan and work off of plans that you have already developed. This is extremely problematic when code you have already written becomes useless with new requirements, or when things that were done are now not even close. Specifically for us, we could see the customer requesting new features that they see are being added on competitor websites that are beyond the original scope.

Well defined requirements at the start of the project help to mitigate these risks, but when the customer needs something you have to accommodate. In order to mitigate these risks you must build time into your schedule to account for requirement changes. This built in time might increase your initial bid costs, but in the end will keep your customer happier.

7. (10 pts) Shortfalls in externally performed tasks

Risk Likelihood – 3

Risk Impact (without mitigation) – 2

Risk Impact (with mitigation) – 1

Shortfalls in externally performed tasks can be extremely dangerous when you rely on external API’s and software that is being created on another team. This is something that you can’t really control, but luckily for us we are mostly doing a visual overhaul so we won’t rely too much on external code, but instead on external artwork. If our art team isn’t able to get us the final products in time it makes all of our work useless.

Once again schedule padding is extremely useful here, if the external workers have a deadline that they don’t meet, but you don’t really need their work until a later date it gives you some wiggle room. It’s still important that your team builds the code up around what you are working on and just leaves out the small external parts that can be added in later.

8. (10 pts) Shortfalls in externally furnished components

Risk Likelihood – 3

Risk Impact (without mitigation) – 2

Risk Impact (with mitigation) – 1

This is the same as the last risk.

9. (10 pts) Real-time performance shortfalls

Risk Likelihood – 2

Risk Impact (without mitigation) – 2

Risk Impact (with mitigation) – 1

Performance shortfalls with web technologies are very common if you don’t follow best practice for traffic and load balancing as well as if you don’t have the server technology to back up your work.

To mitigate these issues we might want to have a network engineering specialist on the team as well as need to express the need for spending on the technology that backs up our work to the customer early in the process so they are okay with the server and technology costs.

10. (10 pts) Straining computer science capabilities

Risk Likelihood – 1

Risk Impact (without mitigation) – 1

Risk Impact (with mitigation) – 1

The field of website design is very well matured and pushing the limits of computer science technology rarely ends up happening because what makes a website good is a responsive user interface and well-designed pages, not pushing the limits of computer science technology by maximizing performance.

This risk mitigation is very simple but relates to gold plating, we don’t need to go beyond the scope of the project and we won’t run into any issues.