**Final Project – Homework 7**

Anish Kunduru

Kellen Johnson

Jonathan Osborne

**Risk Analysis**

1. **Personnel Shortfalls:** Having certain team-members do activities that they have little to no experience in can cause delays as well as cause negative team morale. The best way to minimize this risk is to specifically design jobs for certain individuals to fit their specific strengths.
2. **Unrealistic schedules and budgets**: having certain tasks in the project take too long or unforeseen obstacles when creating the initial schedule can lead to unrealistic schedules that cannot be met. The best way to minimize this risk is to brainstorm and accurately plan for any type of delay that could be caused in the project.
3. **Developing the wrong software functions:** unspecified tasks and functionality in the original mockup of a design can lead to left out or completely wrong functionality in implementation of software. The best way to combat this is to completely analyze, review, finalize, and redo every possible design before continuing with implementation.
4. **Developing the wrong user interface:** unspecified user-abilities and design wants can lead to completely insufficient and irrelevant UI design. If the customer, designers, and users are not on the same page, the user interface result could be confusing, and not helpful to the users. A complete walk-through of how the user should be able to interact with the UI as well as hopeful visual wants should be outlined before the UI is designed. This will minimize unnecessary UI and re-designs due to incomplete/missing features.
5. **Gold Plating:** Implies extra undue effort applied to enhance features by developers. This also includes the addition of new features not requested by the client. While it’s nice for developers to take pride in their work, it can increase the cost and timing of the project. Applying an iterative project management process can keep developers focused on functional requirements. This will allow developers to pitch new features or enhancements to clients.
6. **Continuing stream of requirements change:** Clients and consumers routinely change the scope and design of a web application. This may involve changing the display of the user interface, implementing a new feature, or changing the logic for a specific utility. To mitigate the risk, a development model based on iteration can be used. Development processes such as Spiral and Agile use iteration to implement functionality and verify requirements.
7. **Shortfalls in externally performed tasks:** Our PDM timeline allows for some flexibility to complete tasks late. In addition, having concise and clear needs established with the client, along with regular meetings, will help to mitigate the effect of any shortfalls that are late or inadequately performed. Externally performed tasks involve consumer testing, funding, and other logistical support items.
8. **Shortfalls in externally furnished components** can be avoided through benchmarking, inspections, reference checking, and compatibility analysis. In our project, this is important to make sure that the booking aspect of the website works appropriately. For this reason, we conduct inspections, reference checking, and compatibility analysis to make sure that our backend is compatible with the booking and payments systems.
9. **Real-time performance shortfalls** can be avoided through simulation, benchmarking, modeling, prototyping, instrumentation, and tuning. It is very important to keep an eye out for this top ten risk, since our website is highly focused on the user experience. Unfortunately, this is also a risk that is the hardest to evaluate since it is highly dependent on a user’s hardware and Internet connection. In order to ensure optimal performance, we conduct extensive benchmarking, tuning, and simulations on a wide variety of hardware and software platforms.
10. **Straining computer science capabilities** can be avoided through technical analysis, cost-benefit analysis, prototyping, and reference checking. For our final project, we avoid this through iterative programming techniques (prototyping), technical analysis, and reference checking our pre-existing APIs. Since our main goal is to provide an intuitive and simple UX utilizing external APIs, it should be easy to avoid straining our computer science abilities.