

CSCI 4308 Semester 1 Midterm

Project Charter:

The project charter is a document that is a written agreement between the provider and the consumer. The main sections are the “description and scope”, the “approach”, the “resources”, the “acceptance criteria”, and the “budget.” The “description and scope” describes the problem and the objectives of the project on how the problem will be solved. This allows for clarification on the problem and how the problem plans to be addressed, so that both parties can see whether or not the project is worth doing. In the approach section, we mark major milestones on the schedule, as well as assumptions, constraints, and risk mitigation strategies. However, one of the most important things in this section is that it mentions how to go about changes to the project so that both parties maintain peace. The resources document discusses responsibilities of each member in the provider team, as well as the main stakeholders of the consumers. It also mentions what other resources are needed by the providers in order to complete the project. Acceptance criteria and budget are mostly self explanatory, with the acceptance criteria explaining what defines a successful project and the budget discussing time and cost estimates.

The main purpose of this document is to protect both parties from unexpected changes. This is why a signed charter is necessary. For instance, it means that the customer cannot just randomly expect additional problems to be solved in the project that the providers did not agree to. It also protects the customer, in that they will know whether or not they have received a sufficient solution to the problem, as it is defined explicitly in the contract.

Requirements:

Proper requirements management constricts the scope of the project to prevent scope creep. Scope is best understood as the sum of the requirements. To manage this, it is important to write down requirements and define how changes to the requirements should occur (i.e. voting). High level requirements should be in the project charter and signed off early in the project. Often, high level requirements are broad functional requirements, where we define what an end user should be able to do. Nonfunctional requirements, however, are also important to manage, as they define how the system should behave. Most projects fail because of a failure to manage the requirements, usually due to scope creep and parties changing what needs to be done throughout the project. If this isn't carefully managed, it is easy to lose track of what needs to be done for the project to be “finished.” This is often why projects take longer to accomplish than expected.

Teamwork:

So far in this project, I have learned a lot about teamwork. I now consider the three most critical success factors for working on a project to be compromise, communication, and delegation. For instance, one of our team members has difficulty meeting because he is in an opposite timezone. This is why one of the most important factors to a successful team is compromise. Without compromise, the team would just be unable to meet and unable to identify what we ought to do in order to achieve something, especially when two people have opposing views. Of course, right alongside that is the importance of communication. With insufficient communication, we would be waiting for people to show up to meetings without knowing when, and disagreeing on what something meant in the project. Delegation is also important, as it allows us to complete tasks at a super efficient rate. Without delegation, people get confused and things don't get done.

Lessons Learned:

I have also learned a few important lessons throughout this project. One thing that I have learned is that it is important to have someone on the team doing project management and the majority of the communication with the sponsor. Without that, people get confused on who should answer what, who is in charge of replying to what email, and it often leads to an awkward silence in zoom calls. Obviously, no one should be hogging the conversation, but to have a default person doing the communication avoids problems, and it allows for people to specialize in answering different parts of the application.

Another thing is that it is both a blessing and a curse to have non-technical people in charge of you. Because of having a non-technical sponsor, our team has a lot more freedom in how we achieve the tasks listed in the project, and we can be a lot more creative with the tasks. It also means we don't constantly have someone looking over our shoulder regarding what we are coding. However, it also means that we also sometimes have to explain how some things are done, and what can and can't be done for what reasons. Sometimes this can be difficult.

The last big thing that I have learned is that not knowing about a topic is okay, and that it is okay to specialize in something. Personally, I have very little knowledge of the machine learning part of the application, but a lot of the other people do. I, on the other hand, know a lot about web-scraping, which is not known by most of the other team. Though I want to learn about machine learning, and sometimes I talk with those team members about machine learning, I am mostly concerned with the architecture of web scraping. However, because of this specialization, we are able to accomplish more. I have to trust that these other people can do what I cannot.