CUSTOM FIT CODING, LLC

Project Postmortem

Cooking to Goal

Stephanie Kinsella

Spring 2011

Project Overview

Our group, Custom Fit Coding, worked on a semester long project called Cooking to Goal. Cooking to Goal is an application designed to help its customers achieve their nutritional goals by scaling recipes to meet specific nutritional values, such as total fat, carbohydrates or sugars. Our program allows its users to create a profile or use the program without logging in, add/edit/delete recipes or use pre-entered recipes, print recipes and create/print weekly shopping lists based on the ingredients needed to cook each recipe for the week.

What went right?

The first thing our team did absolutely right was in choosing to create a Google group. The goggle group allowed us to basically set up an email alias that more or less distributed emails to everyone on the groups list. We communicated through this group so that we all knew what was going on with the project status and also utilized Google's "Documents" and "Calendar" features. The Calendar allowed us to each enter our schedules into a shared calendar so that we could plan meeting times and not waste time with useless meetings when the whole group couldn't meet. The Documents feature allowed us to create, upload, and edit documents in real time with the entire group. This enabled us to all see the same document in real time, each edit different sections of it, communicate via the chat section of the document, and not have to worry about emailing out the most recent version of documents. These Google tools were, in my opinion, our biggest asset. It enabled us to communicate effectively in the Distance Education format to compliment the use of the Adobe Connect room.

We used the Adobe Connect "room" that was provided to us by the professor. This is a space where our team was able to meet with audio and video capabilities and share computer screens. This allowed us to work collaboratively to work on deliverables "together", as close as we could get being distributed throughout the state.

Both of these tools in concert with one another truly allowed us to communicate and work together effectively in this format.

What went wrong?

The first thing that we did wrong was that we didn’t follow what we planned to do. We spent the majority of the semester working with the customer to come up with a solid list of requirements that we would deliver in the first iteration and what we would save for future iterations, as well as coming up with a solid design, and in the end when it came to implementation, we did not follow exactly what we planned to do. We ended up letting our scope get out of control and began implementing things that we agreed to wait on. The second thing that we did wrong was that often times we all thought that we were on the same page with how we were going to design and implement things and in the end, when we were close to time for a deliverable during a phase, we came to realize that most of us were on a different page and we needed to backtrack, regroup, refocus.

Risk Management Assessment

In my opinion we managed risk poorly and this whole process was a huge risk. The biggest frustration of the process was going into it knowing the actions and such that could lead projects to failure and falling right into those traps ourselves. I feel that we could have mitigated these risks (such as scope creep) much better. A very large risk that we took was that we did not follow a formal process for merging code or restricting access to our code repository. We allowed all members of the team access to push and pull code from the repository which in the end led to very large problems. This allowed people to overwrite code that others had written and work on portions of code assigned to others. If we had assigned control of the repository to one person, the team leader, and allowed only this person to push and pull code, we would have avoided this problem, because at one point near the end of the implementation phase, we had to take a huge step back because we did not even recognize our code and what we had. Another risk that we took was in the tools that we chose to use throughout the process. We ended up choosing to use Net Beans for our GUI (we used Eclipse for the rest of the coding) as it provided an easier interface for developing the GUI. We also used Git for version control. These two pieces of software along with some other things were completely new to many people in our group. With this said, we not only were responsible for completing the assignment on time and to the best of our ability, but in addition we had to learn these tools. This was a big risk as it put our schedule at risk. At times we were not sure if we were going to finish everything on time. We could have easily chose to not used these “fancy” packages or used things that we already knew how to use, but this is the path we chose to take.

Change Management

I think that our change management was almost non-existent this go-around. We continually had to take a step back and realize that we were trying to implement features that were reserved for future iterations when we hadn’t finished first iteration features yet. We had veered so far from our original design that our code was almost unrecognizable for a while and it took us a while to get back on track which was a big waste of time.

Conclusions

Through thick and thin this group worked together to get things done. One thing we had going for us was that we worked well together. Even though we did not complete in full the scaling function, we were able to include many functions into our application. Overall our process and application was a very big success. This was a huge learning experience for me – this is the very first time I have ever gone through the process of Software Engineering from beginning to end let alone do it with a group. We did a great job of pulling together and doing what we needed to do to make it work, especially in the geographic circumstances we were placed under for this course. If I had to do this project over again I would like to see a strict modified waterfall method utilized with special attention paid to making sure we are on track and implementing according to design. No process can be flawless, that is just not the nature of Software Engineering, so even though surely we made some mistakes, I feel that this was a learning process and we learned from our mistakes and well exceeded the expectations we had for ourselves.