The first group project I ever worked on as a Computer Engineering student at Iowa State University was a disaster. During my second semester, I took the optional one credit class CPR E 186, Introduction to Computer Engineering and Problem Solving II. The course’s only assignment was to come up with an idea for a product of some sort, assemble a group, and spend the rest of the semester developing it with that group.

In the first few weeks of the course, I formed a group, and we came up with a fantastic idea. The idea was to create a software system connected to a custom mechanical device which would take black and white images as input and have the mechanical device draw it on a whiteboard like a printer would on a piece of paper. We didn’t realize how in-depth the idea was, we just thought it would be a cool and fun idea! We had an inkling that it was a large undertaking, but we figured that if we all work hard, we could make this project a reality. We were wrong.

Our final design required understanding in designing mechanical, electrical, and software systems, of which none of us had even an apprentice level of understanding. All the members of our team were getting our first exposure to software systems that semester in COM S 227, I was the only member with experience in 3D modeling, and no one had a clue how we were going to assemble the mechanism we proposed once we had the parts we weren’t even sure we needed.

We also had team management issues. When deciding on meeting times, some of our group members didn’t tell us they weren’t going to be available for half the meeting time. When we did meet, we never set goals for ourselves or achievements we planned to make, ensuring that after the meeting, no one had any expectations for what work they needed to produce by the next meeting.

Finally, our idea for the project was… silly. It was a project that solved no existing problem we could think of, could only be used in a very specific case, and only existed to be flashy. In context, this is a fine project for a group of eager engineering freshmen. However, our idea that the rest of our college experience was going to be about building things we thought we were cool despite no one else having a use for it was especially wrong.

This description of this project and its failures is somewhat long-winded, but I love thinking about it when thinking back on my college career. This project perfectly exemplifies how I’ve grown during my time at Iowa State, wherein I learned the actual systems information I needed to develop the products I wanted, learned how to be a part of a team and help manage the team’s operations, and what it means to be a useful engineer who can work to solve problems on a global scale, not just build contraptions I thinks are cool.

The most important part of becoming an engineer is of course gaining knowledge of systems and designs in your field. I gained most of my understanding of the systems that Computer Engineers interact with through my classes and internships. In COM S 227 and COM S 228, I got a great understanding of the algorithms and data structures that are used everyday to solve all kinds of programming problems, as well as how to apply them in a useful object-oriented fashion. In CPR E 381, I learned the fine details of how a MIPS processor is built by building one. It helped me understand the complexity of what needs to happen under the hood to get my code working, and gave me an appreciation of how much more complex modern processors are. Taking COM S 327 as well as CPR E 308 helped fill in the gap between higher level software and the underlying processor by teaching me proper low-level C programming, as well as some of the details of how an operating system needs to manage hardware resources to make our modern operating systems possible.

Through my internships, I was able to apply much of my knowledge gained through classes and apply it to real life. Through my work in Clearwater Analytics, I learned how to create new software components which can be inserted into existing systems, communicating with backend servers, and how to create helpful documentation so future programmers can better understand your code. These were all lessons first taught in classes such as COM S 228, and solidified in my professional work.

Through those internships, I also learned better group communication. At Clearwater, I would have regular meetings with my team to talk about the state of our projects and the goals we wanted to set. We would then cement those plans by using a project management software, Jira, which would allow us to assign tasks to each other and label priorities to them. This mirrored my experiences in COM S 309, where we were required to use Trello to keep track of our team’s progress. During that time, we also did regular code reviews of our teammate’s code, to help ensure the quality of our implementation, and to keep our code formatting and styling consistent.

Better communication with my group was also essential to success when I did projects with peers outside of school settings, especially for HackISU. During hackathons, it was imperative to be able to talk efficiently with your group so that the most work possible could be crammed into the short 36 hour period. Taking pointers from COM S 309, we would judiciously use Git to share our code and keep track of each other’s updates, so we could remain on the same page as long as possible.

A large part of my transformation as an engineer came during my time spent at HackISU. My first few attendances yielded no rewards from the judges, which was disappointing, but at the time not for us concerning. We were having fun programming stuff with our friends! It wasn’t about impressing anyone. However, we later realized we did want to win, we just didn’t know how. That’s when we started understanding what it means to give your product a purpose.

During my most recent attendance to HackISU, me and my friends designed an app thinking about the end-user first, and having empathy for them. This was an idea introduced to us when we were taking the design thinking portion of our CPR E 288 course. Using the story of a recent fatal attack on a beloved fellow ISU student while she was alone, we were inspired to design an app which would not only attempt to provide security to students in similar situations, but also raise awareness during the hackathon of issues such as these. Sadly, we did not manage to secure any awards when competing next to our very skilled peers. We did however receive much more positive feedback from the judges we talked to.

Sometimes I think about going back to that old whiteboard printer project, to finish what I started in a poetic fashion with my new technical skills, understanding of group management, and by bringing an infusion of purpose to the project. I’m very grateful that ISU gave me the chance to fail that completely, so that I could later on see how much I’ve grown during my time here.