

# Lessons Learned

## Lessons Learned on the Experience with the HCI Process

- 1. Writing out scenarios is a very effective way of putting yourself in the user's shoes.** I think it is very hard for programmers to think as users. We automatically think of how to build something, and I think that almost always sacrifices usability. Before coming up with the right solutions for the user, we already think of how we are supposed to code that piece, which might result in us not building it because we are already aware of the technical difficulties. However, when I was forced to go think through each problem scenario, I was really forced to think like a user, and what problems I might be having. *I started to feel their pain.*

Then, when I wrote the activity scenarios, my eyes were opened about what I, as a user should be able to do. Often times, we go through an application, and we wonder, “*Why can't I do this particular activity that seems so obvious?*” Now I know how to prevent that.

Writing out information and interaction scenarios allowed to me architect the information and to present it without getting overwhelmed. I really like that I now have a framework for coming up with solutions to problems users are actually suffering through. It was very eye-opening to see how tightly intertwined the problem, activity, information, and interaction phases were. As I wrote out a scenario for one, a question comes up that results in my having to alter a previous phase. And then when I made the prototype, that immensely changed a lot of the scenarios. Coming from an Agile programming experience, I really liked that iterative, build on the last one, way of doing things.

- 2. A great user experience is about solving problems.** When I was doing usability testing, nobody said, “Wow this feature is innovative!” but rather, “Wow, I wish they had this when I was looking for a job!” or “Wow, I wish they had this when I was trying to hire people for my startup!” Therefore, I have come to believe that a real breakthrough experience for a customer is about our designing products that solve their problems. We find these problems by interviewing our customers, by going out in the field capturing their tacit knowledge. We further hone in what we think they want solved by putting ourselves in their shoes by writing scenarios, which we should validate early on with things like prototypes and usability tests.
- 3. Validate first (and early) and build later.** In retrospect, I think I should have validated my design approach a lot sooner than I had. When I asked my users to validate my prototype, it was a working, good looking one made in Axure – complete with colors, links, and the like. I think that took away a lot of usability insights because my testers were distracted about the colors and not actual usability concerns such as page

navigation, content, flow. They were telling me things like, this color is not good. Or that the page was boring – like an airline website. Therefore, I think I should have definitely done low fidelity prototyping early on, so that users are fully (even in their subconscious) aware that these are mocks, so that we can all focus on actual usability and not the frivolous 2% of a prototype. Testing early one would have saved me a lot of hours that resulted in redesign.

- 4. Simple is not always better.** We have all heard that simple is better, especially in the lean startup world where we push for minimum viable products. I beg to differ (also, popular software blogger and owner of software company Fog Creek Software – Joel Spolsky always has said that every time they release a new feature, their revenue spikes). When I was doing usability tests, a lot of my testers were asking why I didn't have x feature and y feature. While we have to be careful not to choke the application with too many features, we do have to think if we are designing a solution that allows users to do what they want to do (which goes hand in hand with writing out the scenarios as I have mentioned.)
- 5. But simple things make a big impact.** A lot of my users got confused because of labels such as “engagement”. We often take for granted how little things influence the user's mental model in a big way, so it was great reminder that little things like labels and positioning can impact usability in a positive way for little effort.

## Results of the Prototype Evaluation

The general thoughts on the application itself was wonderful. People were begging me to actually build this application for real. One even asked me if I had quit my job and was founding a startup! The following is an aggregation of the survey results:

- I have to rethink the slideshow. Comments have said that they like the idea (one even said it was their favorite), but others have stated it can seem a little too much. Perhaps this is due to the speed of the transition? Is it too big? Do I need to place it in a different location? These are some things that I will need to bring back to the drawing board and usability test.
- Everyone agreed that finding a job or a candidate was relatively straight forward. That was great news as I was a bit concerned that the filters were too much.
- All liked how they were portrayed in their personal profile. This too was very good news as I really wanted to do away with the linear approach of Linked In.
- All were able to understand the culture of AirBnb, our sample company profile, means the application was able to capture and portray what a user would want to know about a company.

- Not everyone was comfortable about importing data. Like I have mentioned, this is where we really need to examine the tradeoffs and balance between not burdening the user by asking for too much information and invading their privacy.
- It doesn't look like Icarus is seen as a replacement for Linked In, which is a very good thing. I specifically positioned the design to target exactly what Linked In was lacking. We don't want to compete with Linked In. We want to compliment it.
- It looks like we are doing an okay job with reaching and retaining passive candidates, but it could be better. This might improve when I implement the newsletters.

## Summary of the Final Design

The final design resulted in two main views: company view and candidate view.

Company view is all about helping companies find candidates that match their culture. Currently, the design assumes that we are curating these tech and startup companies, so it is not showing quizzes and edit mode of the company view. Once logging in, a person in the company view (say a recruiter), will be taken to a dashboard which has a slideshow, a trending section, a pulse section, and the search section. The slideshow is curated content from the recruiters network with things like articles on recruiting in the tech industry and keeping company culture positive. The trends section is meant to keep recruiters in the know about what is hot in the tech industry and help them position how they look for candidates. The pulse section aggregates all of their social networks into one stream – but filtered to only show professional related content. The search includes search boxes, dropdowns, radio buttons, and checkboxes to help companies hone in the right candidate. Results are shown on the top of the page with percent of matching visible under the name. Clicking on a name brings the recruiter to a candidate profile page which has a video, and innovative sections such as Passions and Projects, Social Media, Essays, and Interview Questions. Most importantly, it displays graphics on the top of the profile which details why the user is a good match, backed up by analytics and supporting data. Clicking on the email button next to the candidate's name allows the recruiter to quickly contact the person. Clicking on the jobs button allows the recruiter to see all the jobs posted for his or her company in one page.

Candidate view is all about helping individuals find their perfect job and company based on their personality, culture, and innate strengths. We gather this data based on other social media that the user has agreed to sync as well as through their answers to questions on the Quizzes page. Once the user logs in, he is taken to his dashboard which is just like that of the company view. Results are shown on the top, but are jobs and roles instead of personal names. Clicking on a job will take the user to a job page that includes editorial content and a video in the day of the life of that job. It also includes traditional content like job descriptions and minimum qualifications.

Clicking on a company name takes the user to a company profile, which has video, essays, and articles that conveys that company's culture. On both the job page and the company profile page, the user will see his or her levels of matching as well as graphics that explains these further. In their own profile the user can edit his or her information, including links to projects, code snippets, and a video, by clicking on the pencil icon near each section. Going back to the dashboard page and playing with search will allow the user to look for more jobs that are matching.

## Suggested Design Revisions

1. **Rethink import data.** Users were really worried about their privacy. I think this even freaked out one user who worried that what if she didn't use that specific social media? I need to find a way to ease the burden of the user having to answer a lot of questions without asking them to compromise their privacy. Perhaps the question is, do users feel burdened by these quiz questions and how much is enough to provide accurate data? I feel like the latter will not be answered until we actually look at the data science behind this application.
2. **Chat feature and question feature.** A user has asked for this feature, wherein I replied that is what the comments section is for. Another asked for a feature to be able to contact the companies directly – with a guaranteed response. I will have to inspect why these users felt that these two features were needed. Sometimes customers don't know what they want and ask for a feature that may not actually solve the problem they want fixed. I need to make sure, before designing these two features, that they are solving a problem. More importantly, I need to make sure they don't solve a problem for one set of users, but cause a problem for another.
3. **Search.** A user suggested putting the search steps in differently modal forms. The reason I hesitated to do this in the first place is that I worried doing that will make the users feel forced that they have to do a certain step. Also, the last step before the actual search button is a preview of the future results. I was worried that once they see the predicted results, they will want to change a few of the filters, and doing that from a wizard/modal form might require too much back and forth. This will be something I will have to do a usability test on, on multiple subject matter experts, to really see what benefits the user the most.