

A Brief Primer on doing an HCI project in CS 545, Human Computer Interaction

Intro

This document briefly describes the steps necessary to complete a project in CS 545, Human Computer Interaction at Stevens Institute of Technology. The key methodology used is User Centered Design. A key slogan when you are stuck on design issues is to, "Ask the User!"

Note this is based on a 14 week course at Stevens. In the real world the time will vary for your project and probably will be longer.

Steps to Success

Use the following steps as a checklist.

1. In the first few weeks select your project and your team. Assess whether the coding for the project can be done in a month. You need a working prototype for User Centered Design. In selecting your team remember you will all share the same final grade, so work hard as a team and consult your professor when you have difficulties.
2. Remember that your goal is to have a coded, working prototype that people can use. You should manage the code in github and I will expect to get an invite to your github or you deliver the code at the end.
3. Choose your E. Choose only 1 of the 5 Es. You should select a measure(s) to prove that it has improved or worsened.
4. Do a competitive analysis of apps or websites that address similar content. You must show proof of that in your final report. This is part of the industry data.
5. Get other industry data from a variety of sources.
6. If there are gaps in your knowledge after 3 & 4 construct a survey and consult the Frary paper and lecture on questionnaire design.
7. Construct a persona that reflects the data of steps 3, 4 and 5. Do not use a need to use your project as part of the personas description.
8. Write a User Story which is a several paragraph story of a user interacting with your web site or app to get a feel for the user experience.
9. Draw a story board that show the flow of using your app or web site. It should show the large steps, e.g., login, configure, select, filter, ..., but avoid too much detail.
10. Construct a low fidelity prototype, usually hand drawn. Test it on 2-3 friends. You play the role of the program providing the appropriate hand drawn screen as the user uses the app. Have them do several tasks and watch what they consider difficult. Ask for

their feedback and, if you have specific design questions, ask them about them after they provide feedback. Remember to report the number of users.

11. Advance to a moderate fidelity prototype using BalsamiQ, Pop or other software for constructing wire frames. Repeat the use interaction of step 9, but they should be able to interact with the system, at least moving from screen to screen by clicking. Have the users do several specific tasks. Again, ask for their feedback and, if you have specific design questions, ask them about them after they provide feedback. Remember to present the number of users you tested.
12. Build your first working prototype reflecting the feedback you received from the low and moderate fidelity prototypes. Construct a few tasks that you will want the user to do. Determine what you would like to observe from the users while using the app. For example, if you are doing an ecommerce web site, the tasks may be: 1) login 2) order a wallet 3) checkout This may include time spent on the app, errors, time spent on every page, ... Have a brief survey after the app. In the survey include specific questions first and more general questions later. There should be 5-10 users participating in each iteration.
13. Change the design according to the feedback and then do step 11 again. Repeat for minimally 1 more iteration. You can use the same users or different users for each iteration but you must indicate which you did. It would be great if this was accomplished by the 11th week of class.
14. Assign one of your team to be in charge of the presentation and follow the presentation template I provided ensuring that it reflects all the points I listed in the scoring sheet I also provided. Common errors:
 - a. not covering all of the reviews, e.g., PAR, laws of simplicity, ... refer to scoring sheet
 - b. not stating the list (zog, Norman, ...) you used for your Heuristic Evaluation
 - c. not understanding that accessibility is accommodating users with disabilities and not access to the program
15. Construct a ~2 minute video of a person using your app/ web site emphasizing the user experience.
16. Answer the question did you improve the targeted E? This should be supported by data.
17. Other common errors:
 - a. When providing graphs with data, the number of users that contributed to that data, the "n" is not given