Human Computer Interaction

Project Brainstorming and Needfinding

Prof. Andrew D. Bagdanov

Dipartimento di Ingegneria dell'Informazione
Università degli Studi di Firenze
andrew.bagdanov AT unifi.it

November 17, 2017

Outline

1. Project overview
2. Project brainstorming
3. The project document
4. Project evaluation
5. Needfinding
6. The way forward
7. Homework

Project overview

General considerations

Projects

- For projects you are encouraged to work in groups of two.
- Projects will be evaluated based on how well the methodologies in the course are followed in developing an interface from idea, through needfinding and prototyping.
- You are strongly encouraged to discuss your projects with me before beginning, and to develop the project idea during the course of the lectures rather than waiting until the end (i.e. start now).

Types of projects envisioned

- Technical: low-level GUI functionality.
- Standard: a well-executed interface implementation.
- Scientific: a study and re-implementation of an advanced topic.

Projects: technical

When users are other programmers

- The technical project is one in which a need is identified in the context of HCI tools themselves.
- The project will be an implementation of some sort, ideally released to the open source community via github at the end.
- Example 1: an implementation of a new type of widget in the Kivy (or other) framework.
- Example 2: a binding of FRP Observables to Kivy UI components.
- Risks: that the project becomes a poorly-executed hacking exercise (*una spippolata*).
- Evaluation: soundness of implementation, appropriateness for distribution and potential interest to community.

Projects: standard

Everything just right

- The standard project should cover the basic stages of needfinding, prototyping, mock-up, and implementation.
- It should results in fully-realized and implemented user interface or user interface components (not necessarily pretty, nor fully "backended").
- Example 1: interfaces to navigate the wildly confusing and unclear forms and procedures for requesting reimbursement for travel expenses; interviews with users (professors and administrative personnel); expense entry interfaces; prototype form generation.
- Example 2: fine-grained (microservices) toolset for ATAF commuters; interviews with commuters of different demographics (age, frequency, etc); identification and prototyping of multiple microservices (e.g. quick overview, where does this line really go?, etc).
- Risks: project gets overwhelmed with "legwork" (e.g. conducting interviews).
- Evaluation: adherence to principles of needfinding and usability analysis, implementation completeness.

Projects: scientific

Bold and innovative

- A scientific project should address a topic of current research by re-implementing aspects of a published paper (e.g. from CHI).
- Example 1: an approach for determining a user's intention to interact on the basis of his posture and movement in the field of view (dataset and approaches selected from literature).
- Example 2: a content suggestion interface that uses user profiling based on Reddit (or other) user ratings (profiling approach selected from literature on user profiling).
- Risks: that is becomes a vague discussion of *massimi sistemi*.
- Evaluation: adherence to published work, quality of implementation, $+\varepsilon$ bonus.

Projects: technical/scientific

The best of both worlds

- Most projects are really a mix of these basic types.
- For those of you who are more hands-on, engineering-oriented, you could think about building a custom input device.
- This could be best on Arduino components, for example.
- [VIDEO DEMO]

Projects: planning

Planning

- Writing good user interfaces is extremely difficult.
- You will be evaluated not only on the interface implementation, but on how well you apply the guidelines covered in the course.
- I am not a believer in setting random, impossible goals for students.
- That is why it is important to *develop the idea and plan for the project during the course* (again, now is a good time to start).
- In this lesson I will propose three projects (one of each type).
- I will also show an example project document today, which can be used as a template for your own project documents.
- Such a document must be written for each project, and it becomes a sort of agreement about the content of each project.

Projects: planning

Evaluation

- One week before the exam date you must submit all source code (or publish it on Github or similar). Part of the exam grade will be based on a code review.
- Also one week before the exam, you must submit a short (10 pages maximum, 5 pages better) report on your project.
- During the exam you will make a short, 20 minute presentation (informally, on laptop) on the project. It should be clear what the contribution of each partner was to the project.
- After the presentation I will ask each project partner individually about aspects of the course and the project.

Project brainstorming

Project brainstorming

- Now I will propose three different projects (one of each type).
- These proposals obviously reflect my own tastes and experiences.
- They are good points of departure, however, to start a discussion on projects for the course.

Proposal: Cross-device Interaction (scientific)

- The idea is to implement some of the functionality from this paper:

  Schmidt, D., Seifert, J., Rukzio, E. and Gellersen, H., 2012, June. "A cross-device interaction style for mobiles and surfaces." In: Proceedings of the Designing Interactive Systems Conference (pp. 318-327). ACM.

- This system uses tabletop touch information along with accelerometer and gyroscope information from mobile devices to associate touch interactions with a mobile device with a specific user's phone.
- The objective of this work is to enable a richer vocabulary of interactions through phone gestures.
- Additionally, virtual interactions like bi-directional file transfers can be initiated in natural ways.
  [MAKE VIDEO GO NOW]

Proposal: UNIFI Mission App (standard)

- Observation: the forms and procedures for requesting authorization for missions and requesting reimbursement for missions at the UNIFI were apparently *designed by psychotic crackheads who don't know how to use MS Word.*
- Some observations:
  1. These forms are not forms. They are word documents, with alignment done with spacing, and cells nothing more than underscores ('_').
  2. They are constantly changing. It is never clear which form to use, or what information to include where.
  3. One is never sure what information to collect while on a mission, and what to declare and how.
  4. Asking for help almost always makes things worse.
  5. There must be a better way.
- In this project, we will devise a smartphone Mission Application that manages the entire mission procedure from authorization request, through the mission itself supporting scanning of receipts and other expense documents, and up to requesting reimbursement.

Proposal: FRP bindings for Kivy (technical)

- In this project we will implement a binding between the an implementation of Functional Reactive Programming and Kivy (this will make more sense after next Monday's lecture).
- We will use as a basis the RxPy implementation of FRP: `https://github.com/ReactiveX/RxPY`
- We will then investigate different methods of integrating RxPy into the Kivy scheduler.
- We will implement selected method(s) according to the Kivy coding style and best practices, including unit tests where appropriate.
- Finally, we will implement a series of example programs showing off the use and features of our FRP in Kivy.
- The finished package, after evaluation, will be published on Github under a Kivy-compatible open source license.

The project document

The project document

- The Project Document is a sort of contract between me and you.
- Contract is the wrong word, because no project goes as planned from A to Z.
- There are always unexpected "surprises" that can't be predicted from the beginning, and sometimes they prevent us from achieving project goals, or they cause us to change the project goals.
- The project document written at the beginning of the project is also a way for me to monitor and ensure the appropriate level of effort required (not too much, not too little).
- Here is a link to an example project document: `https://goo.gl/vpuiQv`
- Note that the project document is a living document, it is expected to change throughout the course of the project.
- Any changes, especially to expected outcomes, should be made explicit in the document with a justification.
- Ideally, the project document will grow organically to become the final project report.

Project evaluation

Project evaluation

- Projects will be evaluated based on how well the methodologies in the course are followed in developing an interface from idea, through needfinding and prototyping.
- Factors contributing to final grade:
  - Innovation: ambitious projects are more risky, and this will be valued in the final evaluation.
  - Independence: if you take charge of the decision making and execute the project with minimal guidance (advice is not guidance).
  - Technical solidity: the quality of your code counts. This doesn't mean bug-free, but rather good adherence to MVC, separation of concerns, and good coding practices.
  - Presentation quality: how well you present your work also counts.
- Remember: your goal in this project is to design, implement, and evaluate an idea related to HCI.

Student evaluation: REMINDER

- Final grades are based on: a selected programming assignment (more on this later), a project, and questions about the course material.

9 CFU:

Project:	80%
Programming assignment:	15%
Questions on course material:	5%

6 CFU:

Project (reduced scope):	90%
Questions on course material:	10%

Student evaluation: Programming Assignment

Programming assignment (9 CFU only)

- A part of the final grade (15%) will be based on the implementation of a programming assignment.
- This assignment will be a small graphical user interface that you will implement individually.
- Your implementation will be evaluated on the basis of how well you apply the patterns, paradigms, and best practices learned during the course.
- About halfway through the course I will announce the possible programming assignments.
- You must choose one of the assignments from the list.
- IMPORTANT: If you work on a team project, the team members MUST choose different programming assignments.
- Two available projects for this year have been published on the course Moodle (one more to come).

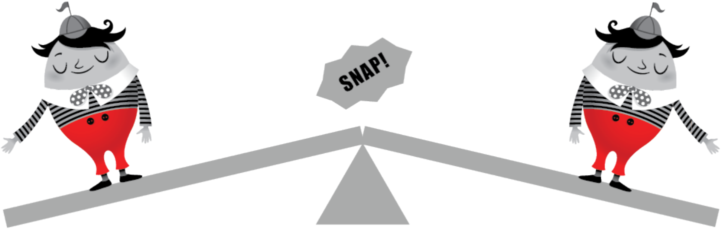
Needfinding

Needfinding

- Needfinding is the process of observing people to discover their needs, goals, and values.
- It is often associated with the process of developing new products or even new businesses.
- The main element is the investment of significant time, effort or money in the development of something new.
- Whether a new product or a new HCI system, it always makes sense to understand whether a genuine need exists.
- A good starting point is to clearly identify an existing problem or need, because finding a big problem and need often yields important untapped opportunities.
- Observing people also helps build empathy and think from their point of view.
- So, how do we observe people and identify their needs?

Needfinding (why?)

The twin anti-poles of design failure



Doing precisely what the user asks

Assuming you know what's best and ignoring the user

Needfinding: participant observation

- It is essential to observe the users and their behavior in context (performing the activity).
- This is vital to learning and understanding their experience.
- While observing, we seek answers to these questions:
    - What do people do now?
    - What values and goals do people have?
    - How are these particular activities embedded in a larger context?
    - What similarities and differences are there across people?
    - Are there any hacks or workarounds used?

Observations

Interviews

Extreme users, lead users

Needfinding: What do people do now?

- Getting a full understanding of the culture, practices, and rituals of your target audience provides an understanding of your audience that is extremely helpful.
- Digging deep into the motivations, emotions, and aspirations of your audience allows you to better understand where to begin.
- Take the UNIFI Forms example, it is essential to understand how real people manage this process today.
- This implies understanding how all types of users (i.e. professors, students, administrative staff, etc) do things.
- Too many software systems are implemented and deployed without a complete understanding of how things are currently managed.
- Without understanding this, it is hard to expect improvement over the *status quo*.

Needfinding: What values and goals?

- Most often, we want to build technologies that align with what people care about and what they hope to accomplish.
- This doesn't mean literally building what people have asked for: people often (usually) don't know how to achieve their goals – especially for disruptive technologies.
- Instead, we must design technologies that will weave themselves into the fabric of everyday life, even if they introduce new concepts and functionality.
- Main point: people cannot be relied upon to tell us how they should accomplish their goals, but through observation and interview we can uncover what those goals are, and what values should be preserved (e.g. saving time).

Needfinding: What is the context?

- For a public transportation user, a bus or subway segment is a part of a larger activity like getting to a friend's house, commuting to work, or going to the grocery store.
- By understanding the constraints and goals of the large activity, we can derive ideas that are otherwise missed if we think narrowly about the bus ride.
- By figuring out why someone would choose to take the bus or not take the bus, we as designers might end up with something more broad, like creating a mobile application that helps people figure out when a bus is coming.
- Taking this broader view can help us be more effective as designers by helping to design for the larger activity that people are engaged in.

Needfinding: Similarities and differences?

- In our bus example, a low mobility user might care about the accessibility of the bus, while somebody else may be concerned with the cost, and yet another with efficiency in getting to hsi destination.
- In the UNIFI Forms example, what similarities might there be between students and professors? Between professors and administrators?
- In many application contexts, there can also be tension between goals and values.
- It is important to understand what these commonalities and differences are in order to develop systems that are genuinely useful to everyone.

Needfinding: Hacks?

- Uncovering hacks that people have discovered for accomplishing tasks is a gold mine for designers.
- This is because hacks represent methods and techniques that accomplish actual goals and respect the values of people in the system.
- Often, these can be translated directly into features in our systems.

Heavy sleeper? Tired of snoozing past your alarm? Use the 'smartphone in a cup' trick to crank the volume!

Needfinding: observing the experienced

- A good strategy for finding needs is to be an apprentice under someone who has experience with the area.
- One illustration of the power of being in the presence of an expert comes from Jack Whalen from Xerox PARC:
  - While studying a call center for photocopier repair, he found that diagnosing photocopiers over the phone is really hard.
  - Unsurprisingly, after studying the support staff they found that the most effective person was the person who had been working there the longest.
  - To their surprise, however, the second most effective person wasn't the person that had been working their the second longest, but the person who had been sitting next to the most effective person.
  - Thus, by sitting next to an expert, these repair technicians were able to pick up informal skills of doing repair work that aren't written down anywhere.

Needfinding: observing the experienced

- If you see something that catches your eye while being an apprentice, be sure to interact and validate it so you can better understand why things are done the way that they are.
- Additionally, pay attention to all the artifacts that compose people's work, because the ways that people have hacked their equipment to make their work more effective is an indication of ways to innovate.
- For example, Post-Its have traditionally been used as artifacts to help users more easily navigate different routines.
- A harmful attitude about designing, although less common nowadays, is the tendency to think people who can't figure out how to use technology are simply incompetent.

Needfinding: observing the experienced

- Using another example from Xerox PARC, Lucy Suchman recorded a video (that has now become legendary) of two people trying to produce a double sided copy of 50 pages of paper:
  - [VIDEO]
  - According to legend, when Lucy shared the video with the executives, they wrote the users off as dumb.
  - However, when it was revealed that the two users were Allen Newell and Ron Kaplan – two of Xerox's premier research scientists, they were no longer able to say that it was because the users did not know how to use technology.
  - Ultimately, this video shows that if you are unfamiliar with a particular piece of technology, it can be difficult to figure out how to use it without an intuitive user interface.

Needfinding: interviewing

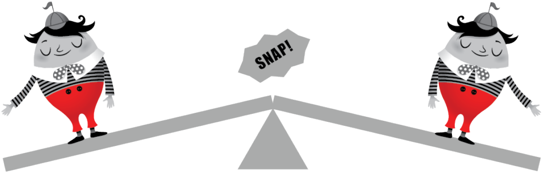
- Of course, apprenticing with a company is probably a luxury for most people.
- Another way of observing people to identify needs is interviewing.
- When it comes to interviewing, one key distinction that must be made is the difference between what people say and what people do.
- For example:
  - Walmart conducted a study asking its customers whether they would like the aisles to be less cluttered.
  - Unsurprisingly, the participants of the study responded: well, yes.
  - Walmart then proceeded to declutter their aisles, remove inventory, and lost a billion dollars in sales.

Needfinding: interviewing

- What happened? In this situation, Walmart made two key mistakes:
  - First, they listened to what people said rather than paying attention to what people did.
  - Second, they asked a leading question.
  - Those two mistakes led them to do exactly the opposite of what would be most effective.

"If I asked people what they wanted, they would have said faster horses" *Henry Ford*

The twin anti-poles of design failure



Doing precisely what the user asks

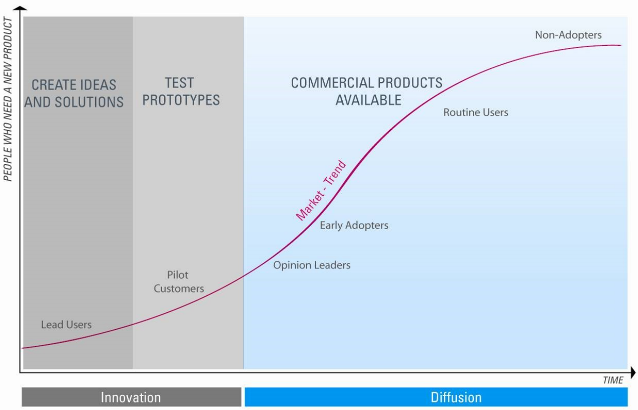
Assuming you know what's best and ignoring the user

Needfinding: interviewing

- The starting point of any good interview is picking a good interviewee.
- A good collection of interviews from people of different backgrounds allows for diverse perspectives that can better help you spot trends that are common among different perspectives.
- If interviewing people for something you are building, you may be able to learn from people who are current users of similar systems.
- Good interviewees can be found through friends, family, and colleagues – however you may not always be able to get an interview with the ideal person.

Needfinding: interviewing

- The key is to be open to any insights that users or potential users may have.
- One group of users, known as lead users, can be extremely helpful in the interview process as well as the development process.
- Lead users are the knowledgeable early adopters that can help provide valuable feedback that directly informs design.



Needfinding: asking good questions

- What makes a good question when interviewing?
- Take this example and decide whether this is a good question:
  - The question is: “Is the daily update an important feature to you?”
  - When asked a question like this, most users would say “Yes.”
  - It’s kind of a leading question.
  - Most people, when asked if something’s important, would say “Sure, why not?”
  - What’s at stake? Why would you say no?
  - If you want to learn about the daily update, participant observation might be a lot more effective – you might even use log files to derive questions.
  - So, for example, you might ask somebody, “I see from the log that you’ve never used the daily update. Why is that? Tell me more.”

Needfinding: asking good questions

- Other kinds of questions to avoid:
  - Questions that ask what users would do, like, or want in hypothetical situations. These questions often generate replies based on a person's ideal, hypothetical world, not necessarily what they would actually do.
  - Questions that ask how often users do things. Asking a user how often they go to the gym generates responses that reveal how often they wish they went to the gym, not actually how often they do.
  - Questions that ask how much they like things on an absolute scale. Asking a user how much they like something is often not extremely valuable or insightful.
  - Binary questions. Questions that only have two answer choices, like questions that ask on an absolute scale do not provide any significant insights into how you should build your product.

Needfinding: asking good questions

- Good questions are questions that are open-ended, have a specific goal, and are unbiased.
- For example, instead of asking how often a person goes to the gym, you should ask them to tell you about the most recent time that they went to the gym.
- Having a specific goal with each question allows you to uncover specific things with each question.
- If a question leaves your interviewee stumped for a little bit, that is a good thing.
- One common problem for new interviewers is that they often say that there's nothing to be found for the problem they're tackling because it's either impossible or obvious.

Needfinding: asking good questions

- However, it's rarely the case that there is nothing new. Malcolm Gladwell has a wonderful explanation in the introduction to his new book of collective stories "What the Dog Saw":

  "The trick to finding ideas is to convince yourself that everyone and everything has a story to tell. I say trick, but what I really mean is challenge, because it's a very hard thing to do. Shampoo doesn't seem interesting? Well, dammit, it must be, and if it isn't, I have to believe that it will ultimately lead me [to something] that is." Malcolm Gladwell

Needfinding: asking good questions

- If an interviewee's answer doesn't quite answer what you were trying to get out of the question, follow up with other questions that clarify what you're trying to get at.
- Paying attention to how your questions are answered also allows you to better understand how to ask good questions that generate the answers you're looking for.
- Examples of good questions:
  - When was the last time you used [some specific application]?
  - What were the first things you did when you woke up this morning?
  - What are three words you would use to describe your experience with [some specific task]?
  - What did you do after a [certain circumstance/situation] happened?

Needfinding: other interviewing aspects

- Another consideration to keep in mind is the location of the interview: they should take place in as realistic of a location as possible.
- For example, interviewing grocery shoppers is best done in grocery stores because it allows the interviewee to answer based on physical triggers that cause different behaviors.
- Conducting interviews in realistic location also allows you as the interviewer to better understand what your interviewee is talking about.

Needfinding: other interviewing aspects

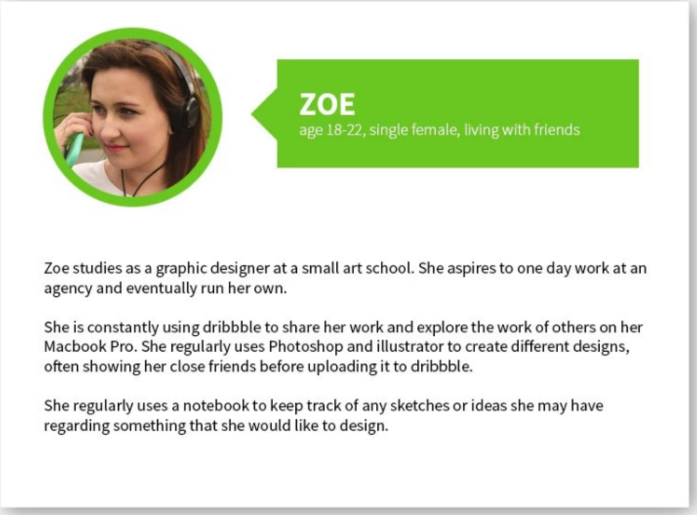
- A common question about interviewing is whether or not it should be recorded.
- Recording audio or video has benefits as well as drawbacks that should be considered.
- The drawbacks of recordings are that they can be time consuming to review and edit, requires permission, but most importantly it can change a participant's response.
- However, recordings provide a robust record and can help you focus on interviewing so you don't have to do that while taking notes.
- A good compromise is taking photos: they are quick and easy to take, provide a visual record, but does so without changing a participant's response.

Needfinding: personas

- The final step, after conducting as many user interviews as you possibly can, is to develop personas.
- Personas are fictional characters that you use to represent the demographics of your users.
- For example, if you are developing a messaging application, your personas can include teenage males who are using your messaging app to communicate with friends, as well as middle-aged mothers who are using the app to message their kids.

Needfinding: personas

- Example persona:



ZOE

age 18-22, single female, living with friends

Zoe studies as a graphic designer at a small art school. She aspires to one day work at an agency and eventually run her own.

She is constantly using dribbble to share her work and explore the work of others on her Macbook Pro. She regularly uses Photoshop and illustrator to create different designs, often showing her close friends before uploading it to dribbble.

She regularly uses a notebook to keep track of any sketches or ideas she may have regarding something that she would like to design.

Needfinding: personas

- The value in having personas is being able to quickly pinpoint different use cases among different demographics of people.
- By assigning a concrete name to a certain demographic, it becomes easier to keep that specific demographic in mind.
- Personas should be as detailed as possible, and be a compilation that represents different groups of your users.
- Beyond just demographic information, however, a persona should also capture a person's motivations, beliefs, intentions, behavior, and goals.
- In other words, give your persona a story to tell.

The way forward

The way forward

- In the next lecture I will give some concrete needfinding examples.
- Then, I will talk about how to take the needfinding process forward into the prototyping phase.
- I will also give an introduction to Functional Reactive Programming, which is an alternative method to managing and modeling events in event-driven programming systems.

Homework

Homework

Exercise 15.1: Zeroing in on a Project

During the week, spend some time thinking about a final project for the course. Try to at least determine:

1. Will it will be an individual or group project?
2. Which type of project: technical, standard, scientific, or some affine combination of the three?
3. What might be a source of users to select interviewees from?