Last time: norms

We act differently in different spaces.

Norms — informal rules that govern behavior — play a massive role in determining how you act in a given space, giving the character to that socio-technical system.

Descriptive and injunctive norms operate differently, but people notice them remarkably quickly, and they are most influential when they are made salient.

Design defaults can influence norms; seeding the community can likewise set expectations.
Where we are, and where we’re going

Weeks 1-2: Basic ingredients — motivation, norms, and starting a community

Coming up:

Friends strong and weak; groups small and large
What happens when it gets big?
I have an idea. Where do I start?

Why “build it and launch” is a bad approach, and how to do better.
Why? Because you never know where things will go...

“He looks like an old retired Tom Holland”

My professor wants us to make a meme for homework, can we make him the meme
When you're an American and nothing else. Definitely not affiliated with Russia at all.

Assignment 1: Going Viral

Recognize how hard it is to do this well, and build intuitions for the challenges and opportunities in social computing design.

Goal: create a piece of content that goes viral.

You must create it. You may remix others' content. Make multiple attempts and iterate! No negativity; create joy, not pain.

Due next Monday at 11:59pm: submit meme to our class server, and submit reflections to Gradescope.

this is a real assignment, just not for a college i attend 😅 i did not think this would get this far

ok anyone reading this: my last tweet is not a northwestern assignment. it's a stanford CS assignment. i'm just trying to do "meta viral" tweet so that nobody else in the course can
Today: getting started

Prototyping

The cold start problem
Social bricolage: prototyping social computing systems
If we’ve learned anything from human-computer interaction (HCI), it’s that it’s a really bad idea to have your idea and then just build it. Just Build It is an immense waste of time, energy and resources in order to find out that your idea was terrible.

Instead of Just Building, HCI pushes for the concept of prototyping.
Prototyping

Progressive fidelity increases as you gain confidence in your idea

Low-fidelity paper prototypes that are rapid to create and explore

Richer instantiations that answer lingering questions

Detailed refinements of the approach

Flickr: Samuel Mann

IDEO

Erin Malone
But traditional low-fidelity prototypes aren’t the right fit here. Why? The basic interface flows aren’t what need to be tested.
Example: On-demand office hours

Imagine that the CS department had allocated a few floating TAs across classes to offer ad-hoc help/support/tutoring on demand. Your idea is to create a social computing system where you can request help, and if any of the floating TAs had previously TA’ed the class you need help with, and are awake, they’ll swing by.

Why is a paper prototype a waste of time here? [1 min]

You’re not really answering the question at the core of the idea.
A prototype answers a question

Prototypes shouldn’t focus on a specific modality, e.g., paper.

Instead, prototypes should focus laserlike on what’s the big risky unanswered question about the idea.

Typically, that question is not whether the interface is usable. It’s how the social dynamics will play out.
Social computing **bricolage**

Don’t build the entire technical stack just to answer a question. Instead, piggyback on existing social computing systems that get you similar affordances. [Grevet and Gilbert 2015]

I call this **social bricolage**: duct taping other systems together so you can layer your social computing design on top to prototype it.

- Push messaging? Use texts.
- Collaboration? Use GDocs or GSlides.
- Chat? Use Slack.

In HCI parlance, social bricolage would be thought of as a specific method for experience prototyping. [Buchenau and Suri 2000]
On-demand office hours II

What’s the big unanswered social dynamics question about the on-demand office hours idea?

How could we use social bricolage to answer it in 24 hours?

[3min]
Example 2: Peer emotional support

Imagine an anonymous social computing system where Stanford students can suggest other Stanford students who need support: conversation, a hug, & etc.

Student volunteers fulfill the request themselves (e.g., go swing by their room), or find and recruit a friend of the target to do it.

No guarantees that this is a good idea. How would you find out?
Prototype simple social formulas

It’s very tempting to try and design out an entire social ecosystem. Don’t.

 Sounds like a Dungeons and Dragons campaign

“Oh we’ll let people post photos of their dogs. If it gets more than 33.5% upvotes, it’ll get a Pupper badge. But then others can sidevote the photos, and if there are enough sidevotes, it starts a chain. Chains are groups of Puppers that are displayed together and aggregate their upvote counts. If someone uploads an old Pupper, the mod (oh by the way there are mods. they’re called Doggos.) can remove the sidevotes and…”
It's very tempting to try and design out an entire social ecosystem. Don't.

“Oh we’ll let people post photos of their dogs. If it gets more than 33.5% upvotes, it’ll get a Pupper badge. But then others can sidevote the photos, and if there are enough sidevotes, it starts a chain. Chains are groups of Puppers that aggregate their upvote counts. If someone uploads an old Pupper, the mod (oh by the way there are mods, they’re called Doggos.) can remove the sidevotes and… Sounds like a Dungeons and Dragons campaign.
Prototype simple social formulas

Instead, identify the simplest set of social interactions that is possible. This focuses the social formula you’re prototyping down to its core.
Don’t prototype sparsely

Typically these systems rely on network effects, like a bunch of friends all using them together. Getting sparse networks by having a few friends from here and there won’t work.

Instead, get a small clique (the graph sense, not the high school sense) to join the prototype together. In other words, a small dense group >>>> a larger diffuse group.
Bootstrapping content and interest

Prototypes do not magically become bustling spaces. Often they feel like ghost towns, because there’s nothing there yet.

When prototyping, you will need to bootstrap the bustling spaces to help set the norms and encourage contributions.

Reddit co-founders initially pushed content until the community took over. (They used many sibyl accounts, which is…ill-advised…)

If it’s a two-sided system — e.g., students and huggers — be prepared to prop up one side of the system to make it useful to the other side.
Diary studies

Mike Krieger of Instagram: early on, running diary studies of Instagram users provided a really useful grounded understanding of what people were actually using the platform to achieve.

Diary study: ask people to keep a structured diary each day of when they used the system, what for, and how they felt.
The cold start problem
Close to 25,000 people had signed up for an account in Instagram’s first 24 hours. The next day, “after getting three hours of sleep,” Systrom recalls, “we were exhausted, but we knew we had created something different. We had a really good feeling about it.”
Cold start problem

The problem: the social computing system isn’t really very enjoyable or useful to anybody when nobody’s there yet. …but then, why would someone join and start populating it, if there’s nobody there?

The entire effort struggles to hit critical mass, like how a car engine on a freezing day can’t start up because it’s too cold, and if it can’t start up, it can’t warm itself up to start. Thus, a cold start problem.
Cold start: how do you hit critical mass?

Critical Mass and Prisoner's Dilemma Problems

Most groupware is only useful if a high percentage of group members use it. Different individuals may choose to use different word processors, but two coauthors must agree to use the same coauthoring tool! Achieving a “critical mass” of users is essential for communication systems [8]. Even one or two defections may cause problems for meeting scheduling, decision support, or project management applications. Even in an idealized situation in which every individual will benefit once critical mass is achieved, the early adopters may well abandon it before the critical mass of users is reached.
Going broad, and trying to get all users at once, typically fails.
Building a focused, engaged core initially is better design.

Why?
- The designer has a clear sense of who the users are, what issues they face, and what norms they expect
- Cliques help overcome the cold start problem
Harvard undergraduates
urban LGBTQ, burners, bloggers
launched as justin.tv, tried to be relevant to everybody, but failed refocused gamers
Affirm my identity

Alex Wolf (bossbabe.com): kickstart by creating a social space that affirms a group’s sense of identity

Look for micro-cultures, e.g., a trend of millennial women graduating from college and not wanting to go the traditional route of subverting their femininity to be professional

At its best, fashion operates similarly: identify micro-cultures that exist and create clothes that affirm that micro-culture’s identity
Close to 25,000 people had signed up for an account in Instagram’s first 24 hours. The next day, “after getting three hours of sleep,” Systrom recalls, “we were exhausted, but we knew we had created something different. We had a really good feeling about it.”
Instagram grew out of a “failed” app called Burbn. Burbn was a mobile check-in app. They realized that people were ignoring the check-ins and just sharing photos, so they started over and built an app just focused on photo sharing.
Identify lead users

Pamela Chen (@NatGeo, then Insta, then HAI, now Apple): one of the first 100 employees at Instagram

On Instagram, one early community was surfers. IRL, there are lots of different kinds of surfers: the best one, the social one, the political one, the one with the surfwear line.

At Instagram, she identified and reached out to the leaders of the community who would be emblematic and shape how other surfers on the site should behave.
Who is the prototypical user?

Starting narrow helps in part because it provides a concrete set of user personas to focus on.

But it’s also important to ask critically: who are we assuming is that prototypical user? Do they look like this?:

Who are we silently assuming is not the prototypical user, and are they often excluded from these dialogues?

Examine the answers to those questions and whether you are comfortable with them.

Hi, I’m a white cis CS guy
As you grow, how do these techniques change?
Are we breaking things?

Prototyping with social bricolage sounds a lot like “Move Fast And Break Things”, Facebook’s motto that has been widely derided because...well, Facebook broke a lot of things.

However, there were a lot of outcomes that would have been fundamentally unknowable from a Harvard dormitory. So, how do we balance it?
Prototyping as annealing

[Scott Klemmer]

Responsibility changes as scale changes:

When Facebook was new and only launched to a few thousand people who knew it was an evolving experiment, I would argue that everything is still very liquid and it’s more reasonable to prototype and iterate rapidly.

However, as the community grows, the system anneals: responsibility grows for addressing the issues created by prior decisions, and for making the next decision more carefully.

Well-publicized reckonings for not doing this quickly enough: Facebook, Wikipedia, Stack Overflow, Instagram, and many more.
Summary

Prototyping social computing systems requires a different approach than usual. Use social bricolage to tie together existing social systems in order to understand the social dynamics you’re creating.

The cold start problem occurs when a system is too empty to attract initial usage, so it remains empty. Two solutions:

- Focus on a narrow group initially, and broaden out later
- Be prepared to bootstrap activity