What was a system you were an early joiner of? How much did it change later?
Group and team collaboration requires interdependence, which leads to a distinct set of design constraints and affordances. Aiming just to replicate the experience of being there is quixotic; better to aim for beyond being there by looking for affordances unique to the digital realm.

Social translucence is a general principle for designing these systems with awareness and accountability.

If incentives are misaligned, these systems will get abandoned.
Where we are, and where we’re going

Week 1-2: Basic ingredients — motivation, norms, and strategies for managing growth

Week 3: Groups small and large

Week 4: Prototyping

Coming up:

Crowdsourcing, anti-social computing, governance and society
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"
Today: getting started

We talked previously about the perils of growth; today we’ll talk about what it takes to get even that far

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

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

Liquid state: Richer instantiations that answer lingering questions

Solid state: detailed refinements of the approach

Flickr: Samuel Mann

IDEO

Erin Malone
But traditional low-fidelity prototypes aren’t the right fit here.

Why?

[2min]

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

For social computing systems, it’s often unnecessary to build the entire social 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.
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.

“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…”
Prototype simple social formulas

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.

<table>
<thead>
<tr>
<th>Beginuur</th>
<th>Einduur</th>
<th>Programme</th>
<th>Wie kijkt mee?</th>
<th>1ste scherm</th>
<th>2de scherm</th>
<th>Welke?</th>
<th>Met wie heb je gecommuniceerd?</th>
<th>Ik voel me</th>
</tr>
</thead>
<tbody>
<tr>
<td>14u</td>
<td>House</td>
<td>niemand</td>
<td>laptop</td>
<td>trein</td>
<td>NEE</td>
<td>niemand</td>
<td>Onplezierig</td>
<td>○○○○○ Plezierig</td>
</tr>
<tr>
<td>15u</td>
<td>Marthe</td>
<td>niemand</td>
<td>laptop</td>
<td>NEE</td>
<td>niemand</td>
<td>niemand</td>
<td>Onplezierig</td>
<td>○○○○○ Plezierig</td>
</tr>
<tr>
<td>20u</td>
<td>Nieuws</td>
<td>Marieke</td>
<td>tv</td>
<td>woonkamer</td>
<td>NEE</td>
<td>mama, ze belde me op de thuistelefoon tijdens het nieuws</td>
<td>○○○○○ Plezierig</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Onplezierig</td>
<td>○○○○○ Plezierig</td>
</tr>
</tbody>
</table>

| Onplezierig | ○○○○○ Plezierig|
| Rustig      | ○○○○○ Actief    |
| Verveeld    | ○○○○○ Geïnspireerd |
| Geïrriteerd | ○○○○○ Tevreden   |
| Geïsoleerd  | ○○○○○ Verbonden  |
|             | ○○○○○ Verbonden  |
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
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.
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.
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.
How do you test new ideas?

How do you A/B test new ideas, when there’s no easy way to bucket people into group A or B? Everyone’s connected…

The most common answer is *country comparisons*, where versions are launched to different countries that have similar properties.

e.g., launch one version in New Zealand and another in Australia

Want an advanced answer? Go chat with Johan Ugander in MS&E:

**Graph Cluster Randomization: Network Exposure to Multiple Universes**

Johan Ugander  Brian Karrer  Lars Backstrom  Jon Kleinberg
How do you build empathy?

How can you build empathy with a huge number of communities? How do you prevent yourself from designing for your own prototypical user?

One approach successfully used in product teams: show user videos to engineers

Bring in stakeholder groups for participatory design

Don’t assume you can. Instead, create local governance (e.g., subreddits) and be responsive to it.
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
Creative Commons images thanks to Kamau Akabueze, Eric Parker, Chris Goldberg, Dick Vos, Wikimedia, MaxPixel.net, Mescon, and Andrew Taylor.

Slide content shareable under a Creative Commons Attribution-NonCommercial 4.0 International License.