



Prototyping and Cold Start

CS 278 | Stanford University | Michael Bernstein



Announcements

Vote on memes by Friday! This is part of Assignment 1.

Project proposal due Weds.: pick a goal, pick a zone, and prototype



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 space, giving the character to that socio-technical system.

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

Design **defaults** can influence norms; **seeding** the community can likewise set expectations.



“Norms” example submitted by Christian A. Jonas

Posted 2013-11-11 by Laila Podlesny

"I need help with this integral:

$$I = \int_{-1}^1 \frac{1}{x} \sqrt{\frac{1+x}{1-x}} \ln\left(\frac{2x^2 + 2x + 1}{2x^2 - 2x + 1}\right) dx.$$

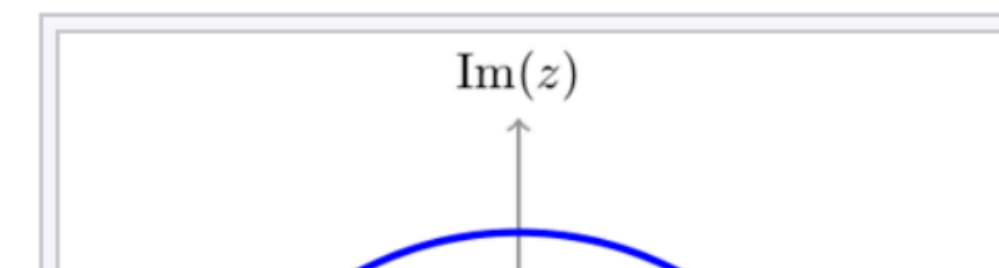
[...] The approximate numeric value of the integral:

$$I = 8.372211626601275661625747121 \dots$$

Neither [Mathematica](#) nor [Maple](#) could find a [closed form](#) for this integral, and lookups of the approximate numeric value in [WolframAlpha](#) and [ISC+](#) did not return plausible closed form candidates either. But I still hope there might be a closed form for it."[\[2\]](#)[\[3\]](#)[\[4\]](#)

The solution was simply stated by Cleo four and a half hours later as:

$$I = 4\pi \operatorname{arccot} \sqrt{\varphi}$$



Norm violation! On Stack Overflow, the norm is to “show your work” so others can follow it. But, a user named Cleo baffled the community by always solving unsolvable integrals within hours—posting just the final (correct) answer, but with no explanation.

0.5% extra credit for examples relevant to recent or upcoming lectures. Submit on Ed under the “Extra Credit” category

Attendance



I have an idea. Where do I start?

Why “just build it” is a bad approach, and how to do better.

Why? Because you never know where things will go...

↑ Posted by u/TncsTurtlJAH 19 1 day ago S
10.5k ↓
My professor wants us to make a meme for homework, can we make him the meme
Meme



when my friends tell me to
stop sending videos of my
professor who looks like
old Tom Holland


TikTok
@danielle_tang



grace :)

he just needs a better haircut😭

3-31 Reply




Today: getting started

Prototyping

The cold start problem

Prototyping social computing systems

HCI 101

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. 

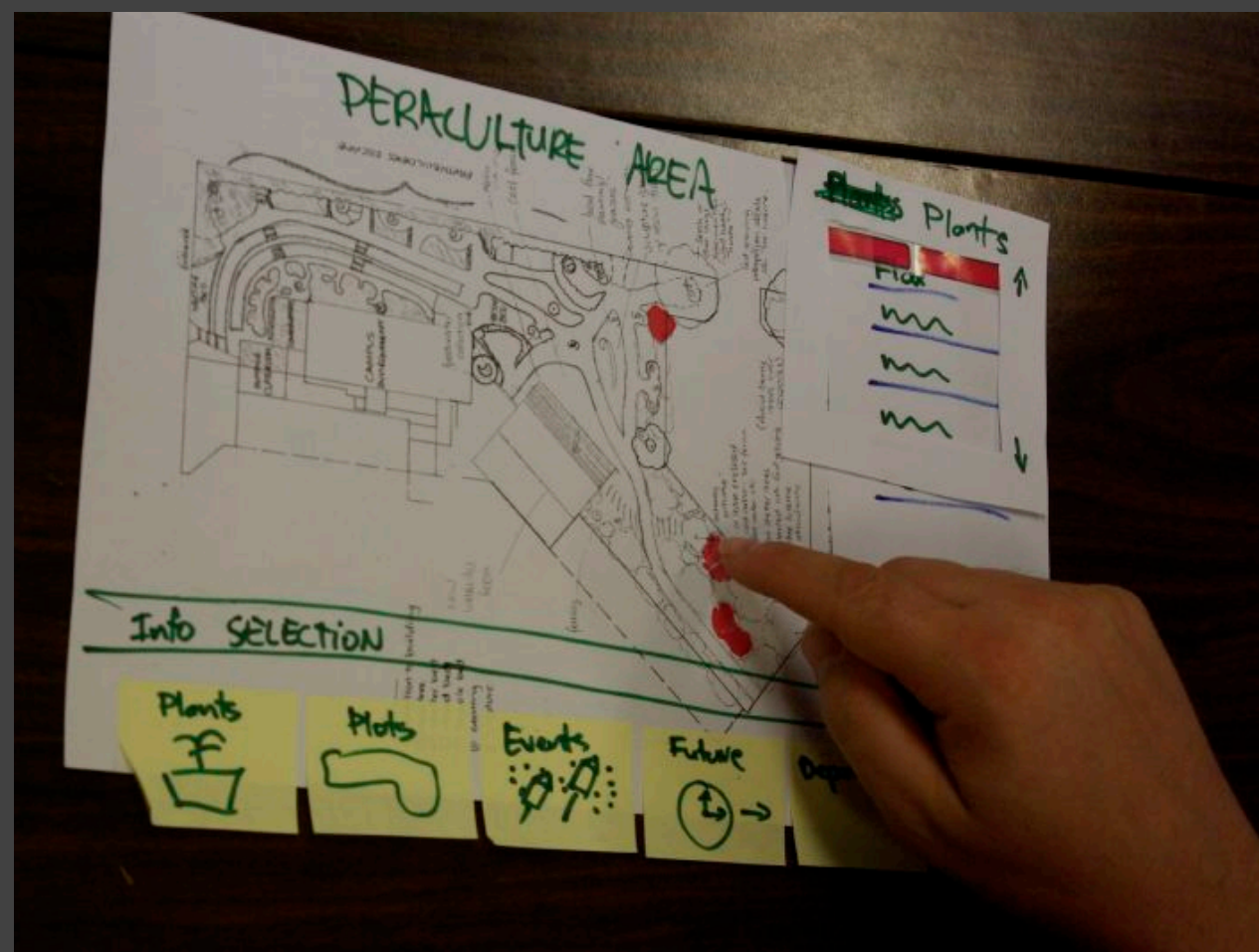
Just Build It is an immense waste of time, energy and resources in order to find out that your idea was terrible.

Instead of Just Build It, HCI pushes for **rapid prototyping**.

Prototyping

Progressive fidelity increases as you gain confidence in your idea

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



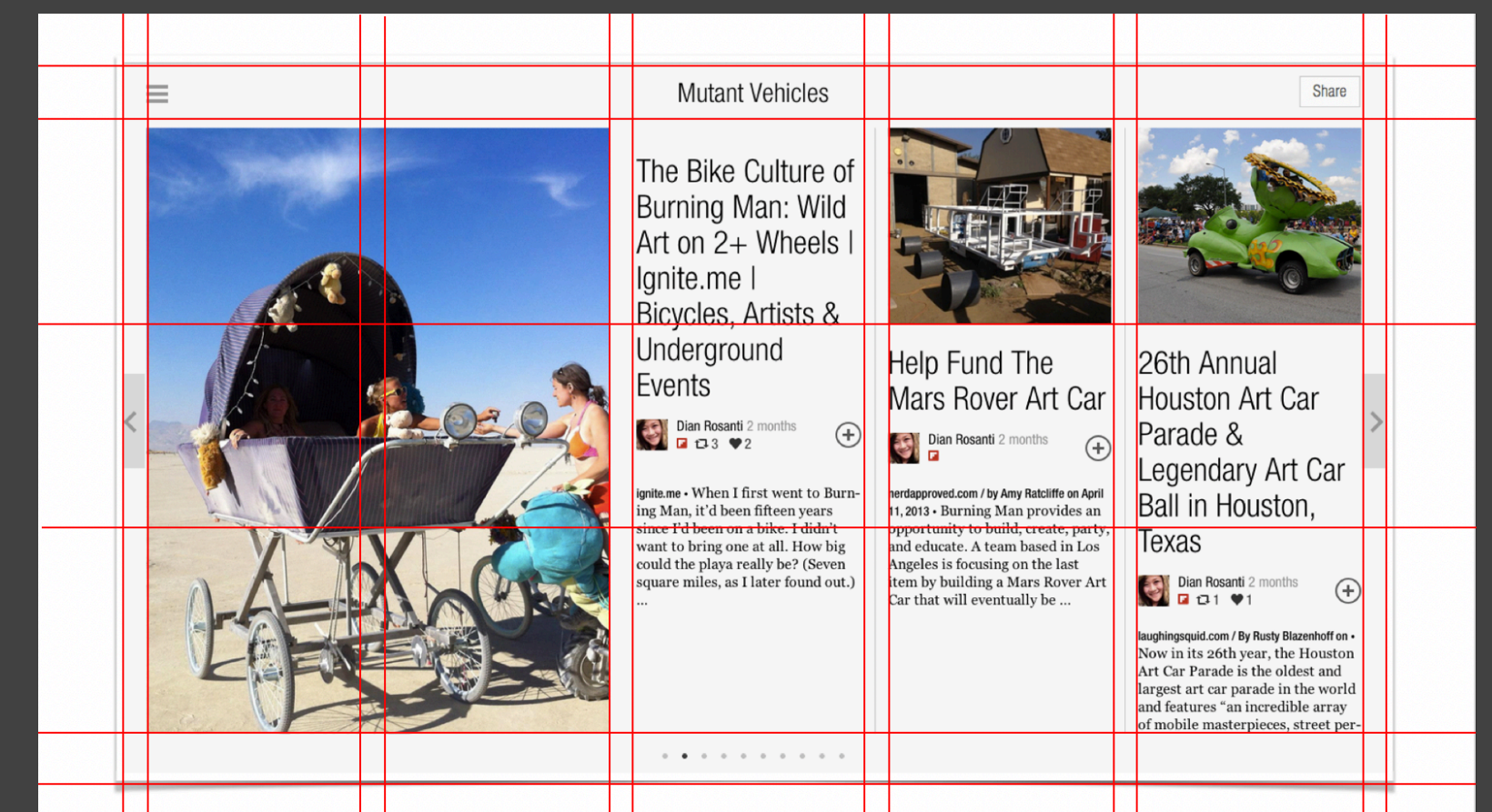
Flickr: Samuel Mann

Richer instantiations that answer lingering questions



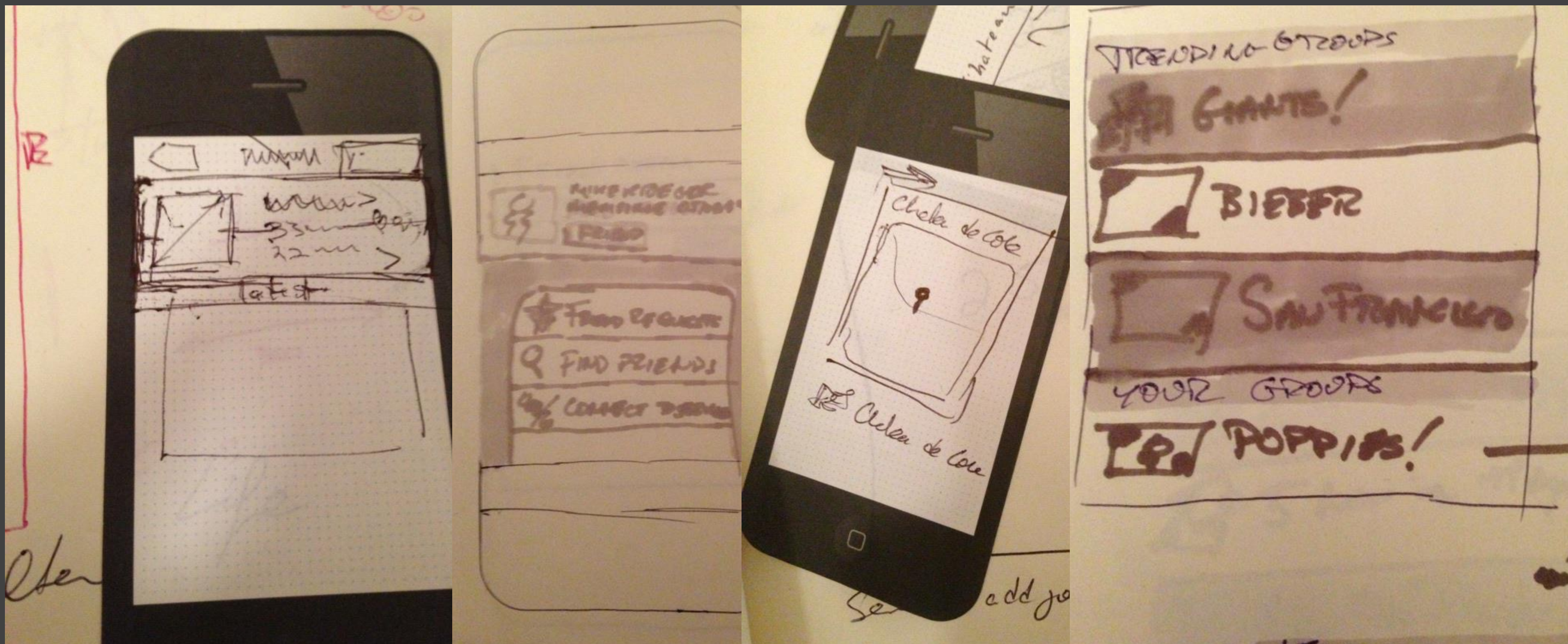
IDEO

Detailed refinements of the approach



Erin Malone

But traditional low-fidelity prototypes aren't the right fit here.



Why?
[1 min]

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?

You're not really answering the question at the core of the idea.

Prototypes should answer a question: **how will the social dynamics play out?**

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.**

Piggyback prototyping

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]

In other words, layer your design on top of existing social platforms to prototype the social interactions.

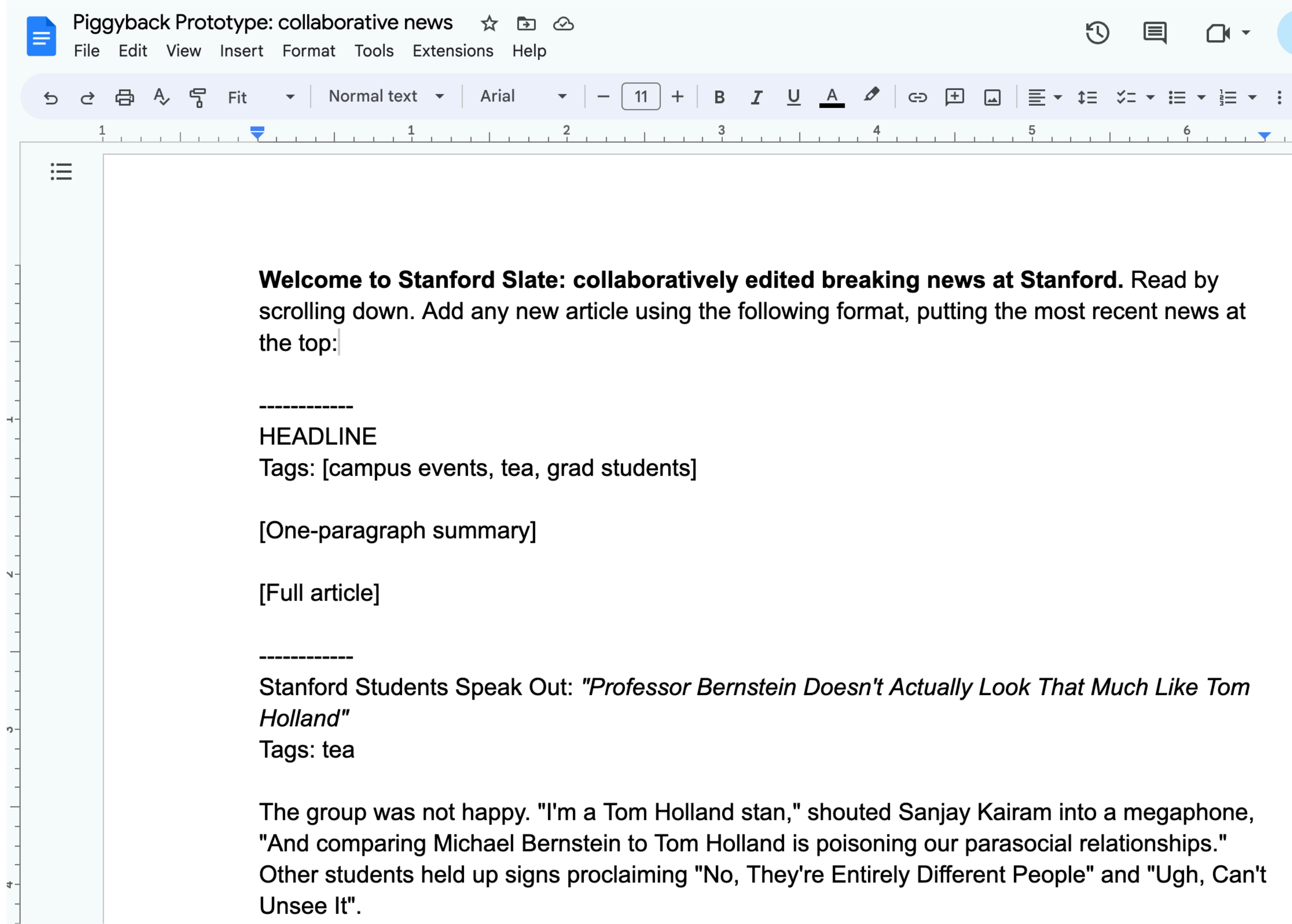
Push messaging? Use texts.

Collaboration? Use GDocs or GSlides.

Chat? Use Slack or GroupMe.

In HCI parlance, piggyback prototyping can be thought of as a specific method for experience prototyping. [Buchenau and Suri 2000]

Layer
structure
into the
existing tool
to guide
behavior



On-demand office hours II

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

How could we use piggyback prototyping to answer it in 24 hours?

[2min]

Pick the piggyback platform

Local news, but with
Wikipedia-style open
contributions

Communities for
tracking growth in
personal skills (e.g.,
fitness, practicing music)

Your new Twitter clone

Slack / Discord
Google Docs
YouTube
Group Chat

(There is room for
more than one
reasonable approach for
each of these. Focus on
which platform provides
the best affordances for
testing the social
behaviors.)

Pick the piggyback platform

Local news, but with
Wikipedia-style open
contributions

Communities for
tracking growth in
personal skills (e.g.,
fitness, practicing music)

Your new Twitter clone

Slack / Discord
Google Docs
YouTube
Group Chat

Create a Google Doc
to track a specific topic
for a week and invite
folks to contribute

Pick the piggyback platform

Local news, but with
Wikipedia-style open
contributions

Communities for
tracking growth in
personal skills (e.g.,
fitness, practicing music)

Your new Twitter clone

Slack / Discord
Google Docs
YouTube
Group Chat

Sign up for a goal you're
interested in, populate
channels by goal, and post
updates regularly to that
channel

Pick the piggyback platform

Local news, but with
Wikipedia-style open
contributions

Communities for
tracking growth in
personal skills (e.g.,
fitness, practicing music)

Your new Twitter clone

Slack / Discord
Google Docs
YouTube
Group Chat

I “post” tweets to the
group chat, and others
can react

Approaches that are **not** piggyback prototyping

Not piggyback prototyping: running a survey to ask “would you use this system?” or “what would you do if you saw this interface?”

(Besides not getting an authentic sense of what the social behavior will be, asking questions like this is also unlikely to give you good data. Take CS 347 to learn why.)

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...

Prototy

It's very tempti
Don't.

“Oh we'll let
upvotes, it'll g
photos, and
groups of Pu
counts. If som
are mods. th

My zombie apocalypse team

Fighter



Sniper



Zombie bait



Insane guy



Driver



Navigator



Most likely to survive



Leader



mulas

al ecosystem.
ragons campaign
more than 33.5%
evote the
n. Chains are
gate their upvote
by the way there
tes and...

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.

Prototype “the hard side”

[Chen 2022]

Think critically: whose contributions will be the most challenging to recruit?

in a student+faculty forum, it's hardest to get faculty to reply

in tiktok or wikipedia, it's the content creators

in an art feedback forum, it's: the artists, or the feedback givers?

My take: it's the feedback givers: they are the scarce resource

Nearly every system has different roles: always prototype the role representing “the hard side” — if that part fails, the whole thing fails

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., interns and mentors — be prepared to prop up one side of the system to make it useful to the other side.

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."

This is an outlier.

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?

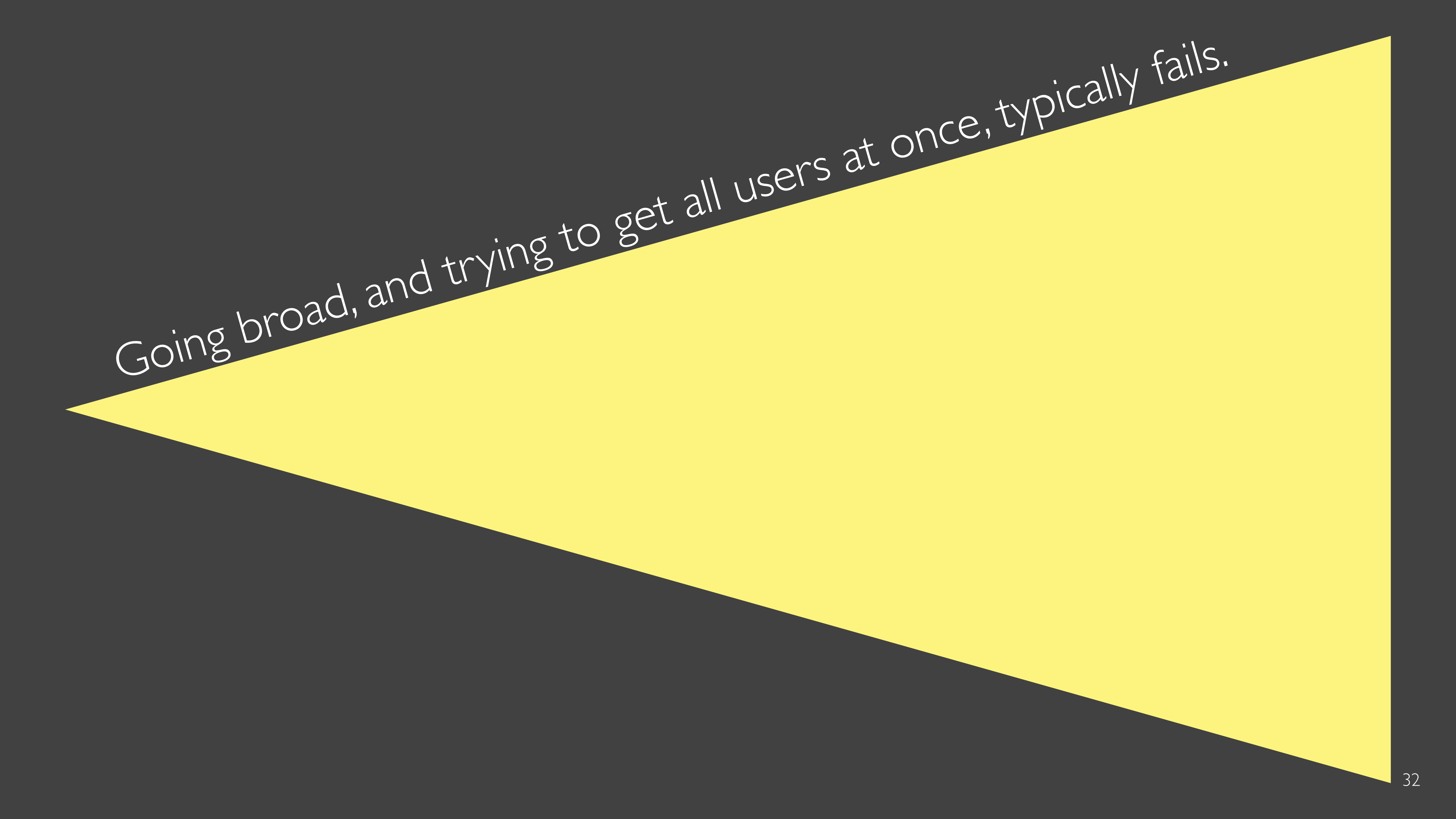
The entire effort struggles to hit critical mass, like how a car engine on a freezing day can't start up because it's too cold, and if it can't start up, it can't warm itself up to start. Thus, a **cold start** problem.

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.

[Grudin 1998]



Going broad, and trying to get all users at once, typically fails.

Building a focused, engaged core initially is better design.

Why?

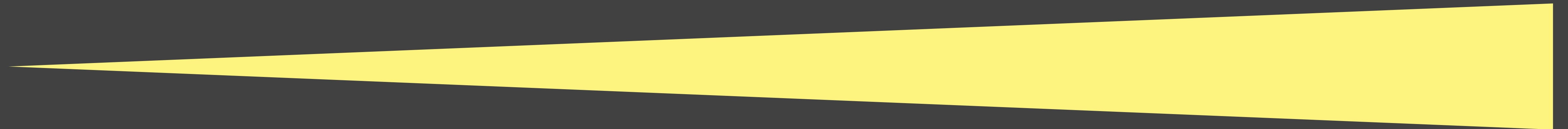
- The designer has a clear sense of who the users are, what issues they face, and what norms they expect
- Cliques help overcome the cold start problem
- This an “atomic network”: aim to create the smallest stable, self-sustaining group possible [Chen 2022]



Harvard
undergraduates

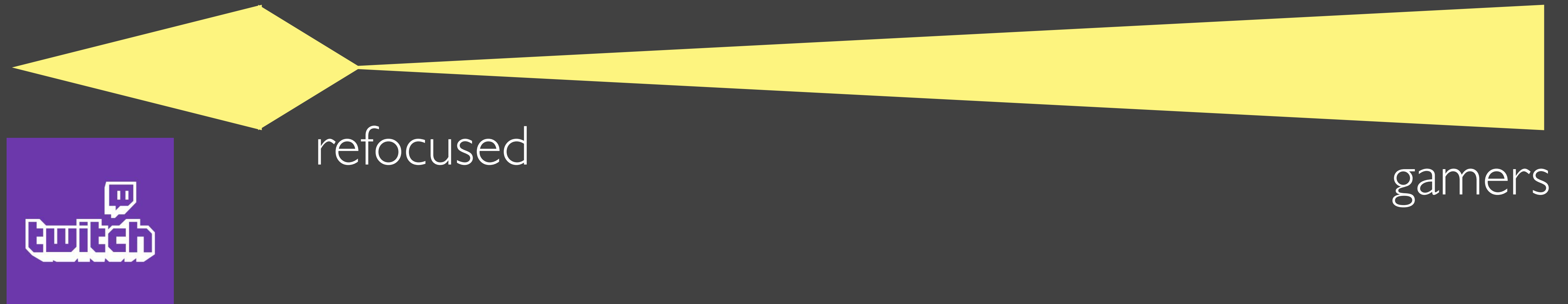


handicraft meetups at local boutiques



food delivery on Stanford campus

launched as justin.tv,
tried to be relevant to
everybody, but failed



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

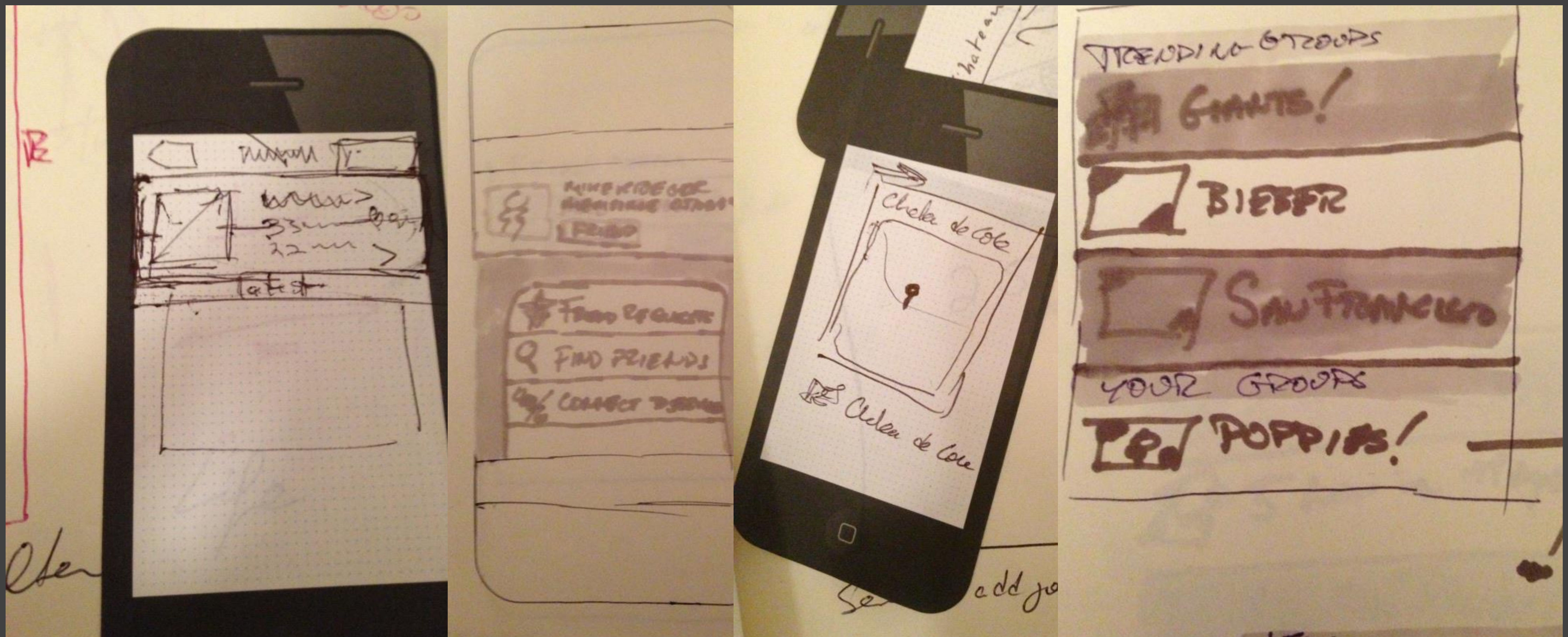
At its best, **fashion** operates similarly: identify micro-cultures that exist and create clothes that affirm that micro-culture's identity





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."

~~This is an outlier.~~
This is a lie.



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



@pc

Who is the assumed 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?

Reflect on these questions before you prototype.

Hi, I'm a
white cis
CS guy



**As you grow, how do
these techniques
change?**

Are we breaking things?

Prototyping with piggyback prototyping 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?

One view: 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 was still very liquid, and it's more reasonable to prototype and iterate rapidly

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

Summary

Prototyping social computing systems requires a different approach than usual. Use piggyback prototyping to tie together existing social systems **with the goal of testing the social dynamics**.

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

References

Buchenuau, Marion, and Jane Fulton Suri. "Experience prototyping." Proceedings of the 3rd conference on Designing interactive systems: processes, practices, methods, and techniques. 2000.

Chen, Andrew. 2022. The Cold Start Problem, Harper Business.

Epstein, Daniel A., et al. "Revisiting Piggyback Prototyping: Examining Benefits and Tradeoffs in Extending Existing Social Computing Systems." *Proceedings of the ACM on Human-Computer Interaction* 6.CSCW2 (2022): 1-28.

Grevet, Catherine, and Eric Gilbert. "Piggyback prototyping: Using existing, large-scale social computing systems to prototype new ones." Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems. 2015.

Grudin, Jonathan. "Groupware and social dynamics: Eight challenges for developers." *Communications of the ACM* 37.1 (1994): 92-105.

Social Computing

CS 278 | Stanford University | Michael Bernstein

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.