Pokemon Go designs for strong ties by encouraging consistent interaction with your friends in the game. For every day that you send or receive a gift from your friend, you’ll grow your friendship level. With higher friendship level milestones, you get in-game rewards and other bonuses.

“Strong and Weak Ties” example submitted by Matthew Y.

0.5% extra credit for examples relevant to recent or upcoming lectures. Submit on Ed under the “Lectures” category
Collaboration
CS 278 | Stanford University | Michael Bernstein
Announcements

Assignment 2 is due Tuesday

Project milestone will be due the Wednesday of Week 7

Zone 1: either the front-end or the back-end is functional

Zone 2: no-code components of the project should be complete; code components may still be in progress

Zone 3: launch! no-code components are complete, so take the system live and start recruiting
Last time

Strong ties: a small number of people we know well — design for honest signals, and don’t assume all communication happens through the system.

Weak ties: a large number of acquaintances — design to support feelings of connectedness, but remember that many social systems will be dominated in volume by weak ties.

The impacts of social media use on our wellbeing are most negative for those passively consuming content (esp. from weak ties), and for heavy usage among early adolescent girls.

Today, a different kind of group: one brought together by shared purpose and goal.
Oh #@&%, It Got Popular

Unit 2
We Work

Unit 3
Which team is more effective?

Colocated team has: a room

Distributed team has: Zoom, Slack, Trello, Dropbox, GitHub, Asana, Google Docs, Jira

2:1 more effective
[Olson and Olson 2000; Espinosa 2011; Björn 2014; Hu et al. 2022]

Why? And what can we do about it?
How can we work together?
How can we work together?
COORDINATION NEGLECT: HOW LAY THEORIES OF ORGANIZING COMPLICATE COORDINATION IN ORGANIZATIONS

The team scaling fallacy: Underestimating the declining efficiency of larger teams

Who’s in Charge Here? How Team Authority Structure Shapes Team Leadership

Team Familiarity, Role Experience, and Performance: Evidence from Indian Software Services

The Mutual Knowledge Problem and Its Consequences for Dispersed Collaboration

Out of Sight, Out of Sync: Understanding Conflict in Distributed Teams

The Influence of Shared Mental Models on Team Process and Performance

Some unintended consequences of job design

How might computing augment us in working together?
What tools do we use?

- Dropbox
- Google Docs
- Slack
- Figma
- Gmail
- GitHub
- Zoom
- SharePoint
- Google Calendar

Others?

What design patterns make them successful? [2min]
Today

How do we design tools for effective remote collaboration?

Topics

- Beyond being there
- Social translucence
- Grudin’s paradox
- Remote work
Beyond being there
Goal: being there

Our main goal is to increase fidelity: to try and make the channel have increased richness, allowing for more and more social cues. [Daft and Lengel 1986]

Let's make Zoom and FaceTime have lower delays, higher resolution, and 3D VR or AR scenes.

Let's make coding collaboration tools as effective as if we were pair programming.
Beyond being there

[Hollan and Stornetta 1992]

“Being there” is the wrong goal.

We will never fully recreate the face-to-face experience. There are too many subtle cues for us to fully model or recreate them, even with hypothetical future technology.

Network lag, immersion and comfort issues in VR, lack of shared physical context, …

So, stop trying.
Beyond being there

[Hollan and Stornetta 1992]

Instead of tilting at windmills to design experiences that are as good as being there, design for beyond being there: experiences that could never have been created face-to-face.

How could remote video bring you closer in ways that face-to-face collaboration never could?

How could online coordination tools help us be more effective planners than we ever could with whiteboards and gantt charts?
Examples

Skype translating between languages in real-time and producing foreign language speech in your own voice

Tools that help teams quickly identify if they should be flat or hierarchical, encouraging or critical, and enforcing equal turn-taking [Zhou, Valentine and Bernstein 2018]

Finding just the right person to answer the hard question you are facing, immediately [McDonald and Ackerman 2000]

What are some collaborative superpowers you have or could have?
Social translucence
Awareness [Dourish and Bellotti 1992]

Design must allow people to understand each others’ state and coordinate accordingly, to coordinate interdependencies. This goal is typically achieved through the design pattern of awareness: visualization of others’ activities.

Google Docs

Messaging apps

GitHub

Slack

Significant Otter [Liu et al. 2021]
But awareness can go too far

You don’t want collaborators to know everything…

Whether you’re working at every moment
Draft emails you wrote when you were angry but didn’t send
Dumb bugs that you introduced into your code but fixed quickly before you made a git commit

So how do we walk this line?
Social translucence

[Erickson and Kellogg 2000]

Aim for **socially translucent systems**: give enough information to let natural social cues take over.

Opaque systems: no information
- Solid door to a trafficked stairwell
- Door-in-the-face situation

Transparent systems: total information
- Glass door to a trafficked stairwell
- Everybody feels awkward

Less transparency  More transparency
Social translucence

[Erickson and Kellogg 2000]

Aim for **socially translucent systems**: give enough information to let natural social cues take over.

<table>
<thead>
<tr>
<th>Opaque systems: no information</th>
<th>Translucent systems</th>
<th>Transparent systems: total information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solid door to a trafficked stairwell</td>
<td>Windowed door</td>
<td>Glass door to a trafficked stairwell</td>
</tr>
<tr>
<td>Door-in-the-face situation</td>
<td>Social cues prevail</td>
<td>Everybody feels awkward</td>
</tr>
</tbody>
</table>
Social translucence: example

Erickson and Kellogg 2000

Aim for socially translucent systems: give enough information to let natural social cues take over.

Opaque systems: no information
Code isn’t pushed yet...

More transparency

Less transparency

Transparent systems: total information
Michael Bernstein is editing importantfile.py. He’s typing I am stupid I am stupid I am stupid I am stupid over and over into his code editor.
Social translucence: example

[Erickson and Kellogg 2000]

Aim for **socially translucent systems**: give enough information to let natural social cues take over.

Opaque systems: no information
Code isn’t pushed yet...

Translucent systems
Michael is working on *importantfile.py*

Transparent systems: total information
Michael Bernstein is editing *importantfile.py*. He’s typing *I am stupid I am stupid I am stupid I am stupid* over and over into his code editor.
Social translucence

[Erickson and Kellogg 2000]

Two requirements for social translucence:

1) **Awareness**: others’ activity can be seen — to an extent

2) **Accountability**: others know that their activity can be seen

If done correctly, social translucence supports interdependent work while maintaining plausible deniability when necessary.

If there's no plausible deniability in the system, people will abandon it.
Google Wave: Imagine if the person receiving your email could see all your drafts as you compose them, like the email thread were a Google Doc.
Grudin’s paradox
Why do so many collaborative software systems get abandoned?

Dead wikis and documentation at work

Calendars not reflecting actual person or room availability

“Oh, I don’t use that. Just send me a text instead.”

…even though these systems may even provide social translucence and go beyond being there.
Grudin’s paradox [Grudin 1994]

The socio-technical system may be benefiting everyone…except the people who are expected to use it.

What is in the product manager’s interests may not be in the ordinary users’ interests. [Halverson and Ackerman 2003]

Examples:

The manager wants everybody’s calendars to be up-to-date…but the programmers don’t care, and just want to work on the project.

We want an API to be documented and kept up-to-date, but the people who write and actively use the software don’t need the documentation.

Being on Slack is distracting for the people who need to be reached
Grudin’s paradox [Grudin 1994]

When a system falls prey to Grudin’s paradox, it gets abandoned or circumvented.

How to avoid this? The system needs to provide benefit to all users, not asymmetric benefits.

…And not just perfunctory benefit — enough benefit to justify the work and distraction that using the system might entail.
Hate ‘em, then love ‘em

Irene Greif, who founded the field — and was the first woman to earn a PhD in CS from MIT — spent much of her career in industry research labs working on collaboration tools.

She notes that with each new generation of collaboration technology, companies are extremely wary: all they can see are the risks and the lawsuits. Even with something as simple as voicemail!

Collaboration benefits are much harder to quantify and put into dollar amounts, to balance against the risk. Only later do companies see the value and buy in.
So where are we going?

Meta Horizon Workrooms: VR remote conversations

Using today’s concepts: will this succeed? [2min]
So where are we going?

Beam: robot telepresence robot

Using today’s concepts: will this succeed? [2min]
Michael’s take

All the tools that we talked about today take the organizational structures as given: the team, the teams, the hierarchy, and so on.

e.g., Skype already assumes the members of the team are set

My opinion: the important technologies from here on out will help aid the authoring and evolution of these structures more directly.

Who can be working with who? And how?

What’s the best way for this team to be working together?

Can we recover if we get into conflict and fracture?
Remote work
Back to the remote team...

Colocated team

- has: a room

Distributed team

- has: Zoom, Slack, Trello, Dropbox, GitHub, Asana, Google Docs, Jira

Why? Under what conditions?

2:1 more effective
[Olson and Olson 2000; Espinosa 2011; Björn 2014; Hu et al. 2022]
Remote & hybrid work

Remote work does not have a negative effect on individual execution outcomes

Productivity outcomes go up [Bloom et al. 2015], possibly due to 40% of saved commute time being redirected to work [Aksoy et al. 2023].

Remote work has a negative effect on creative and social outcomes

Firm-wide remote work makes collaboration networks more static and siloed [Yang et al. 2021], makes it less likely that teams find solutions to hidden profile tasks [Javalagi et al. 2023], reduces the creativity of ideas generated [Brucks and Levav 2022], and is associated with fewer breakthrough ideas [Lin, Frey, and Wu 2023].
Yes, even today.

Even as improved remote work tools have made collaboration smoother within teams, they paradoxically make coordination worse across teams [Hu et al. 2022].

Example: what if Stanford forced everyone to use the same Slack/GroupMe/iMessage platform for your projects?
Summary

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.
References


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.