Open Project Management

from an “open” perspective

UNIT 3

Instructor: Dr. Bradly Alicea

http://bradly-alicea.weebly.com

IS 340: Spring 2023

All content

CC BY

Welcome!

A Final Example of Technical Debt in a Community Resource

DevoLearn

Accelerate research in developmental biology with deep learning models

4 followers · Worldwide · http://devoworm.weebly.com

Follow

Overview | Repositories 17 | Projects 1 | Packages | Teams | People 8 | Settings

Pinned

Order updated. Customize pins

devolearn Public
Accelerate data driven research on embryos with deep learning models
Python · 36 · 29

data-science-demos Public
Jupyter Notebook · 1 · 10

Community-Board Public

DevoGraph Public
Jupyter Notebook · 1

nucleus-segmentor_HuggingFace Public
Devolearn Nucleus segmentor model hosted HuggingFace spaces
Python

Synthetic-Embryo Public

Repositories

Find a repository... | Type | Language | Sort | New

View as: **Public**

You are viewing the README and pinned repositories as a public user.

You can create a README file visible to anyone.

Get started with tasks that most successful organizations complete.

Discussions

Set up discussions to engage with your community!

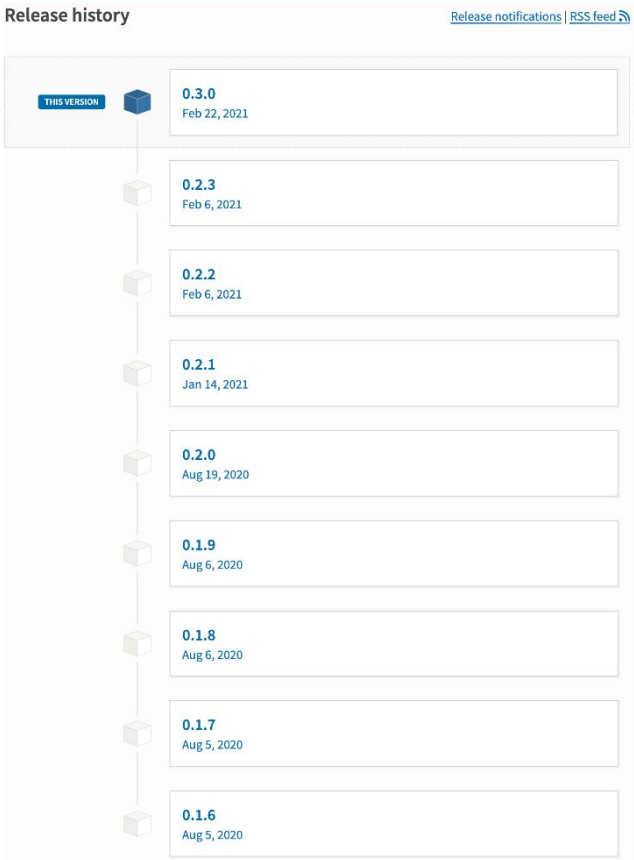
Turn on discussions

People

https://github.com/DevoLearn

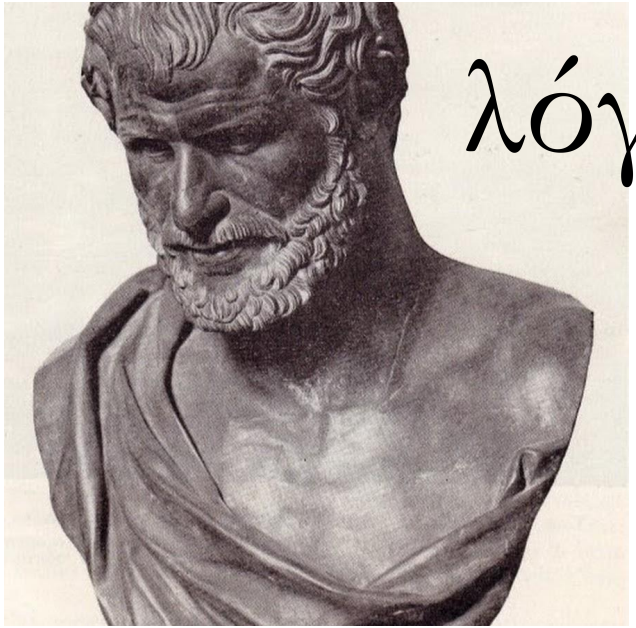
Release schedule is out of sync with Github repository, version of Python is out of sync with project code.

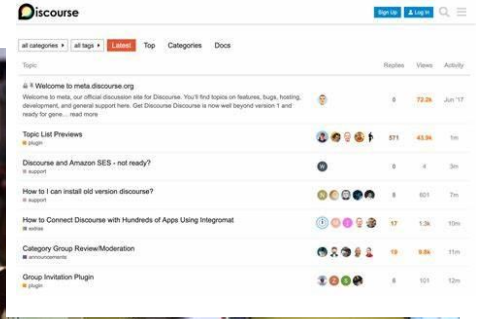
We will return to this in Unit 5.



https://pypistats.org/packages/devolearn

Discussions, Documentation, and Public Events

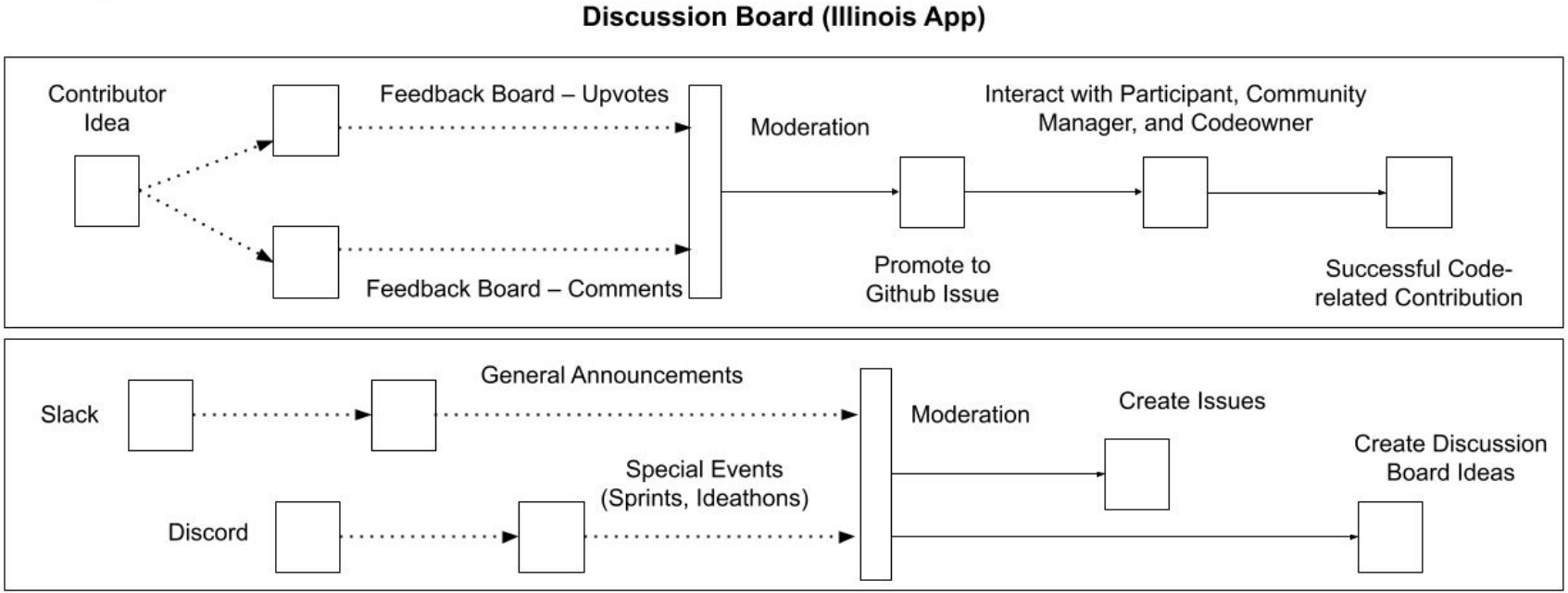




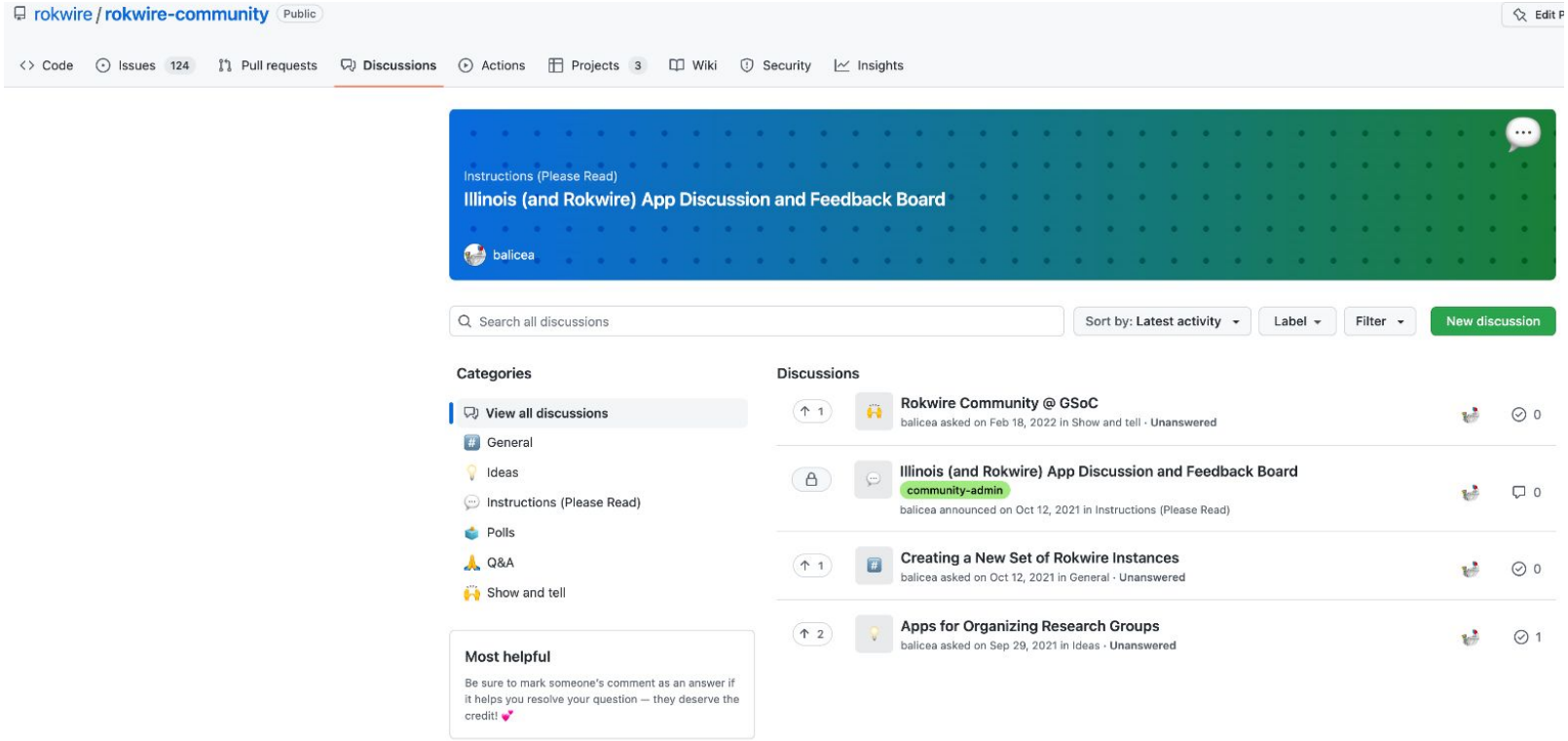
How to Develop Desirable, Useful, and Sustainable Discussion Forums

Participation Type: Leader/Shaper

Becoming a Discussion Leader



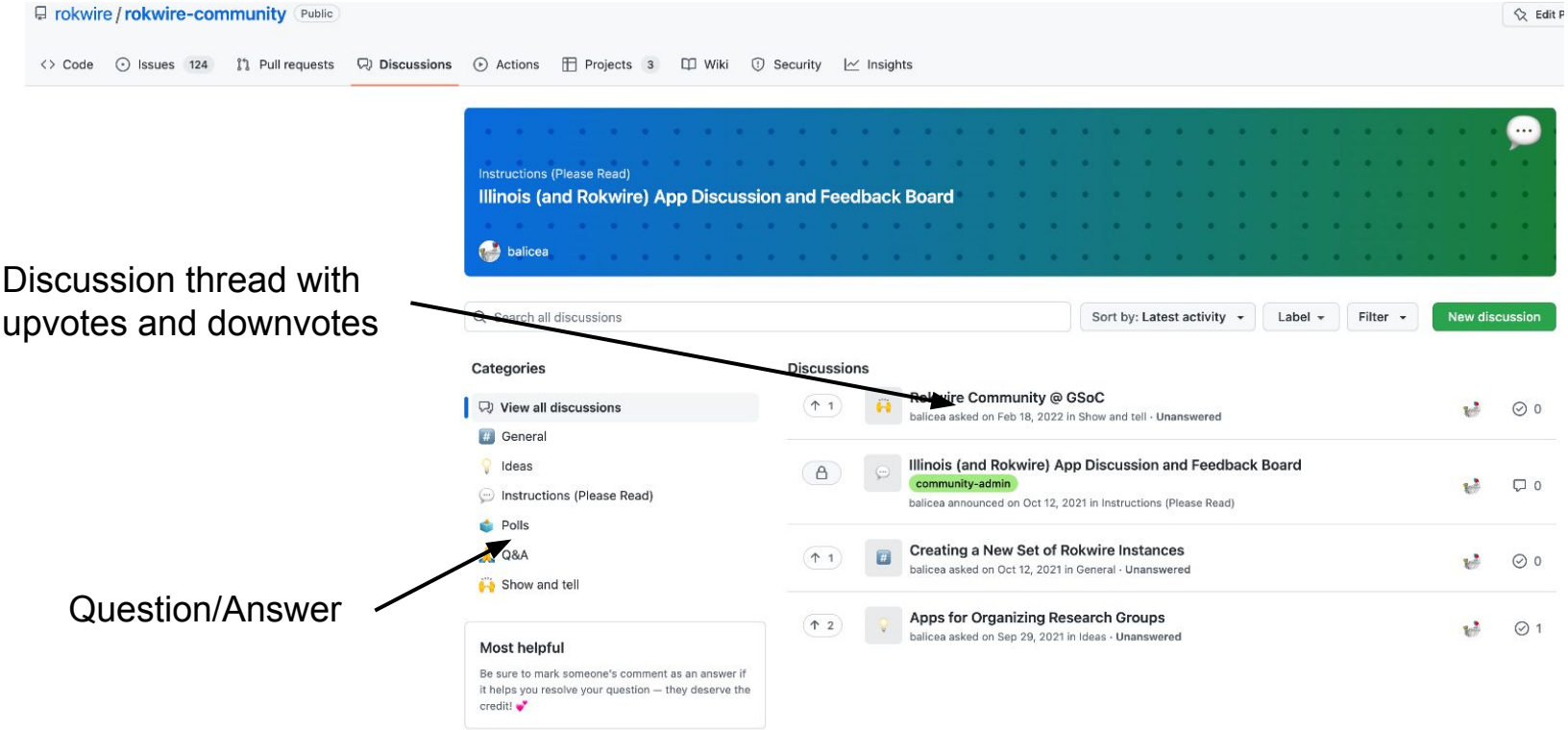
Github Discussions (Feedback Board)



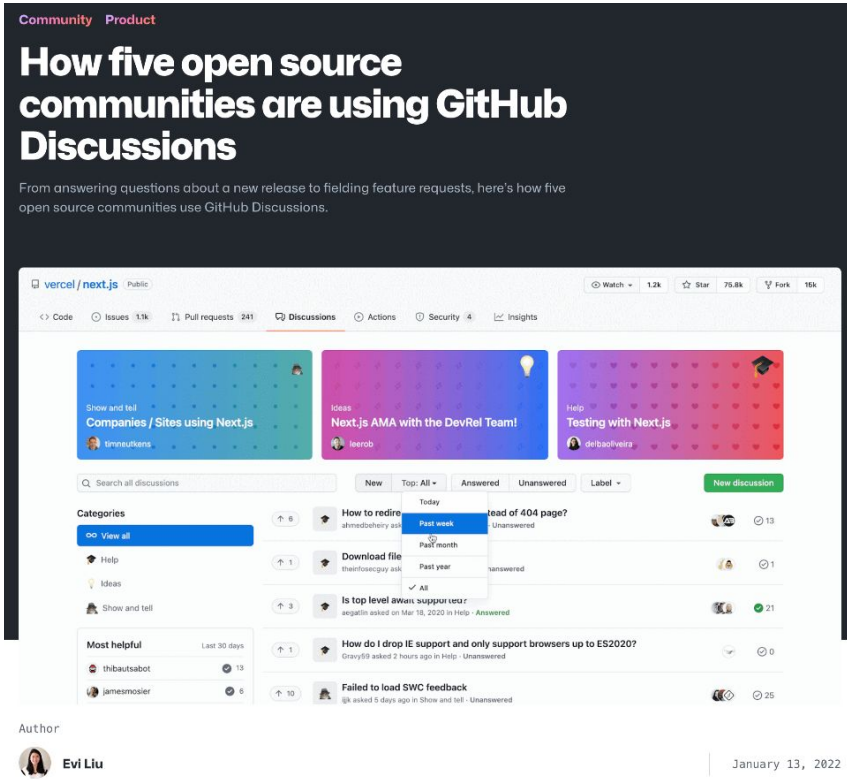
Github Discussions (Feedback Board)

Discussion thread with upvotes and downvotes

Question/Answer



Github Discussions (Feedback Board, con't)



Answering questions after a major release (React18).

Getting feature requests (Dogecoin).

Core flight system development (NASA).

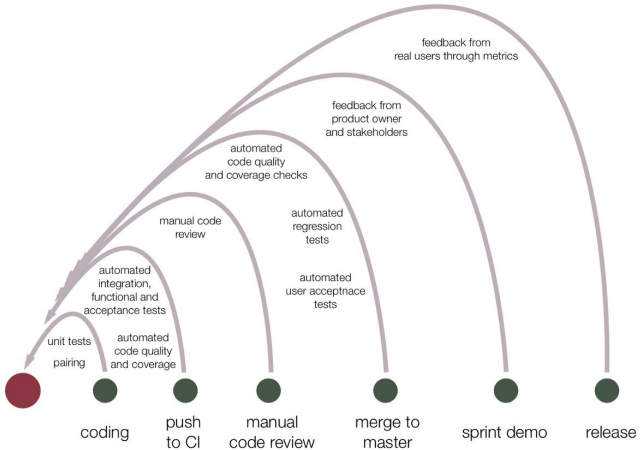
Ordered Timelines for Media (Open Timeline IO - Pixar).

All Phases of Development (Next.js).

https://github.blog/2022-01-13-how-five-open-source-communities-are-using-github-discussions/

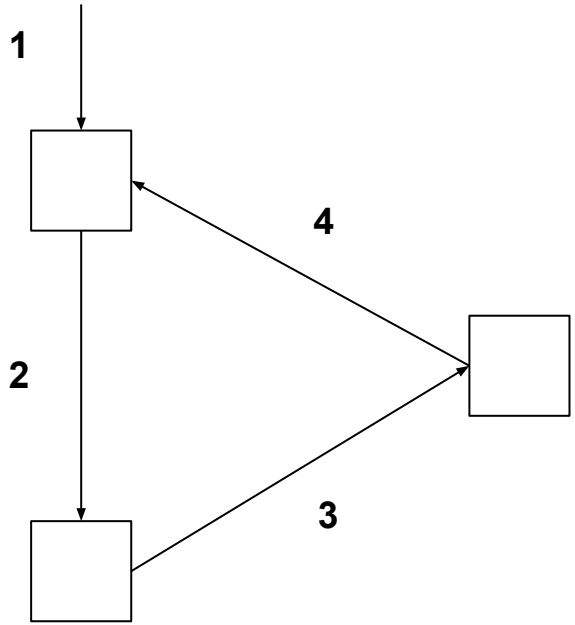
What Are Feedback Loops and Why You Need to Implement Them

https://kanbanize.com/blog/feedback-loops/



“Feedback is a central pillar of Agile. Integrating feedback loops into your agile project workflow ensures you will collect fast and frequent feedback from your customers and increases your chances to adapt to emerging changes”.

Four-step Feedback Loop in Agile



1) capture and store the input – this is the data with which we operate every day.

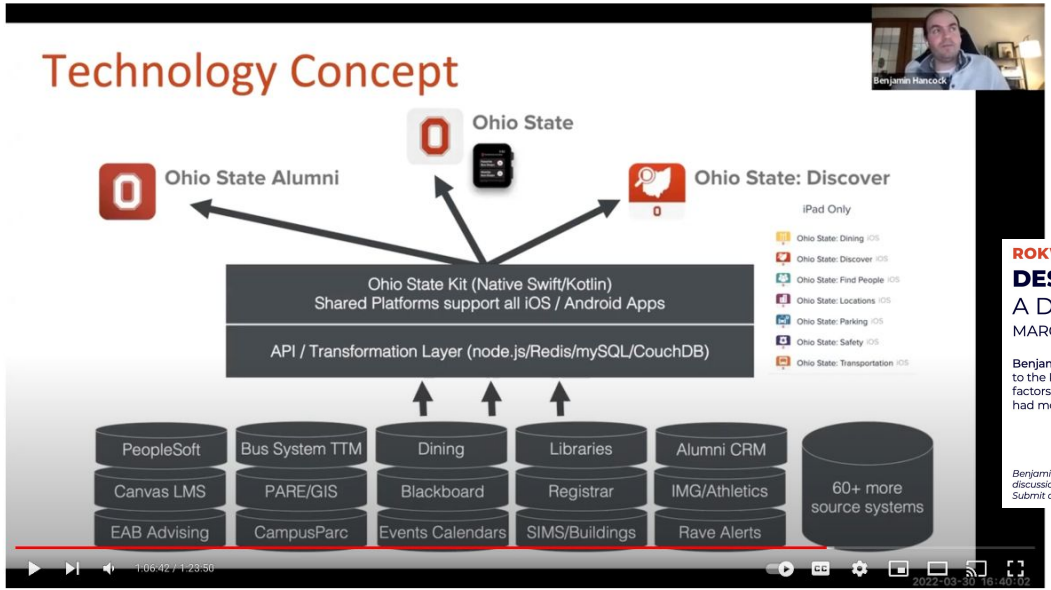
2) analyze the input, gathered from stage one.

3) make decisions based on the insights gained from the second stage.

4) the implementation of the changes we have set forward.

Integrating Discussions with Community Events

Monthly Substack Post on Mobile App Design



Rokwire Forum with Benjamin Hancock: Designing the Ohio State app

ROKWIRE
18 subscribers
Subscribe

0 Share Download Clip Save

39 views 10 months ago
Benjamin Hancock is the Director of Application Development at The Ohio State University. He spoke to the Rokwire team about the design and development of the Ohio State campus mobile app and the factors that guide their decisions. The Ohio State app was launched in 2013. In 2021 the Ohio State app had more than 247 thousand unique users and more than 65 million screen views. Show more

Start with a Special Event with relevance to the community.

ROKWIRE FORUM SERIES

DESIGNING THE OHIO STATE CAMPUS APP

A DISCUSSION WITH BENJAMIN HANCOCK

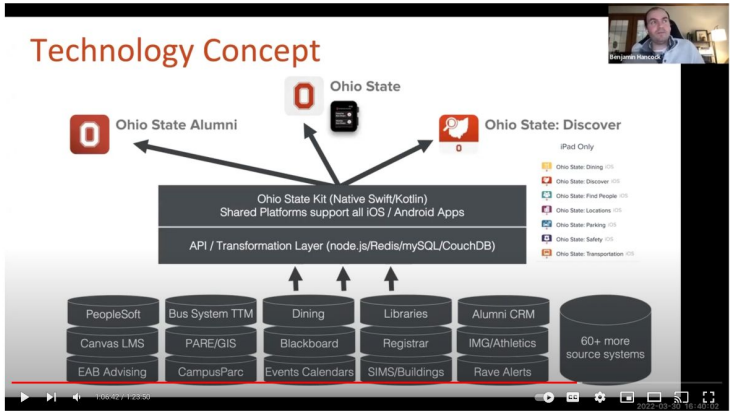
MARCH 30 AT 3:30PM-5:00 ON ZOOM

Benjamin Hancock is the Director of Application Development at The Ohio State University. He will speak to the Rokwire team about the design and development of the Ohio State campus mobile app and the factors that guide their decisions. The Ohio State app was launched in 2013. In 2021 the Ohio State app had more than 247 thousand unique users and more than 65 million screen views.

Benjamin Hancock will be interviewed by William Sullivan, followed by discussion where attendees are encouraged to ask questions.
Submit questions beforehand to Bradly Alicea at balicea@illinois.edu

https://www.youtube.com/watch?v=9Q1P4RIPW3A

Monthly Substack Post on Mobile App Design (con't)



Rokwire Forum with Benjamin Hancock: Designing the Ohio State app

ROKWIRE
18 subscribers
Subscribe

0 Share Download Clip Save

39 views 10 months ago

Benjamin Hancock is the Director of Application Development at The Ohio State University. He spoke to the Rokwire team about the design and development of the Ohio State campus mobile app and the factors that guide their decisions. The Ohio State app was launched in 2013. In 2021 the Ohio State app had more than 247 thousand unique users and more than 65 million screen views. Show more

One big takeaway from the event: success emerges from *doing a few things well*. What kinds of things are engaged with most often? If you are able to make those things work seamlessly, then it is much easier to drive adoption. The Ohio State App has focused on a few core features. Features are added by recognizing a gap and addressing it rather than reinventing the wheel or duplicating an existing app.

Another takeaway is the focus on *reducing friction* in both use and design. Reducing friction means simplifying design elements, making things run smoothly, and quickly allaying user concerns. One example of reducing friction is the notion that *obvious always wins*. In The Ohio State App (below), each button provides enough information for the user to know what it does and whether it is worth their time to click on it. Reducing friction also involves keeping complexity to a minimum, so that a user only encounters essential information. It may be that your minimalist focus reduces the size of your user base a bit, but this also provides a tradeoff of less friction and a better app experience.

Newsletter report on "Campus App Design Principles" that accompanies a YouTube lecture. Summarize and educate on key concepts and issues.

Monthly Substack Post on Mobile App Design (con't)

ROKWIRE FORUM SERIES

DESIGNING THE OHIO STATE CAMPUS APP

A DISCUSSION WITH BENJAMIN HANCOCK

MARCH 30 AT 3:30PM-5:00 ON ZOOM

Benjamin Hancock is the Director of Application Development at The Ohio State University. He will speak to the Rokwire team about the design and development of the Ohio State campus mobile app and the factors that guide their decisions. The Ohio State app was launched in 2013. In 2021 the Ohio State app had more than 247 thousand unique users and more than 65 million screen views.

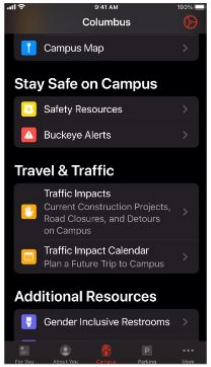
Benjamin Hancock will be interviewed by William Sullivan, followed by discussion where attendees are encouraged to ask questions.
Submit questions beforehand to Bradly Alicea at balicea@illinois.edu

One big takeaway from the event: success emerges from *doing a few things well*. What kinds of things are engaged with most often? If you are able to make those things work seamlessly, then it is much easier to drive adoption. The Ohio State App has focused on a few core features. Features are added by recognizing a gap and addressing it rather than reinventing the wheel or duplicating an existing app.

Another takeaway is the focus on *reducing friction* in both use and design. Reducing friction means simplifying design elements, making things run smoothly, and quickly allaying user concerns. One example of reducing friction is the notion that *obvious always wins*. In The Ohio State App (below), each button provides enough information for the user to know what it does and whether it is worth their time to click on it. Reducing friction also involves keeping complexity to a minimum, so that a user only encounters essential information. It may be that your minimalist focus reduces the size of your user base a bit, but this also provides a tradeoff of less friction and a better app experience.

Newsletter report on "Campus App Design Principles" that accompanies a YouTube lecture. Summarize and educate on key concepts and issues.

Monthly Substack Post on Mobile App Design (con't)



The principles of doing a few things well and reducing friction also intersect in interesting ways. For example, new functionality should reduce the friction potentially encountered by users in the course of trying to achieve their goals. In The Ohio State App, features consist of activities core to the student experience, such as bus schedule, grades, account balance, school calendar. In integrating Canvas, only the features that reduced friction most, such as assignment schedules, can be accessed using The Ohio State App.

Continuous design is the key to longevity, guided by the notion that *design is never done*. This involves a long-term commitment to design that goes beyond the latest feature or release. The Ohio State App has been around for over 10 years, with many design changes over that time. Benjamin has graciously shared a sample design lab in addition to some infographics on annual app usage (from 2019 and 2021).

Expose community to new tools and perspectives, suggest possible new features and directions.

Stream-Feed Integration

Flows of contributor and development activity

- human interactions
- diffusion of skills and code
- task completion.

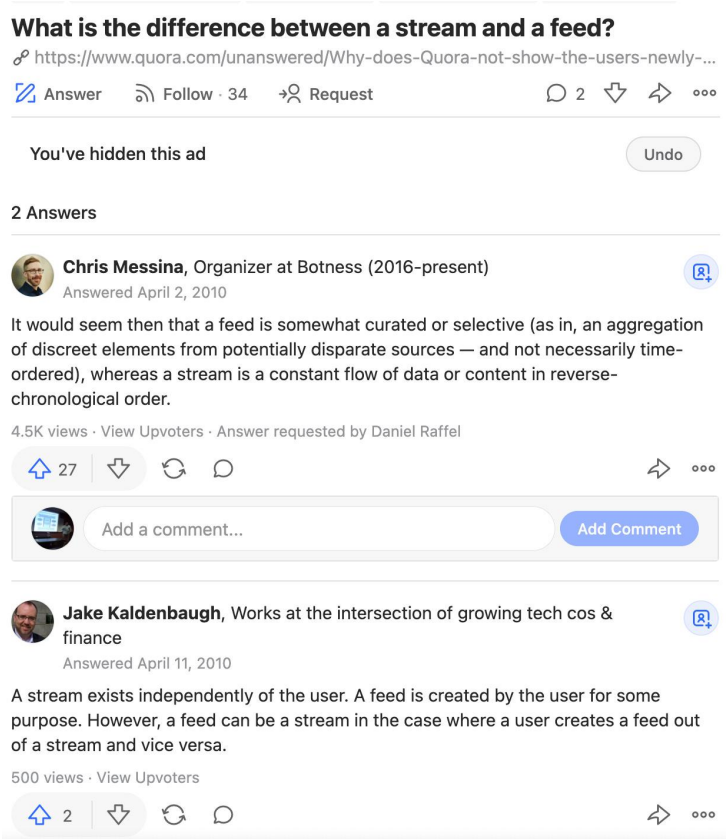
Stream-Feed Integration

Flows of contributor and development activity

- human interactions
- diffusion of skills and code
- task completion.

Flow of media that enables access to short-term and long-term information

- streams are continuous real-time flows of information defined by sources such as live video.
- feeds are curated streams of content from those same sources



Stream-Feed Integration (con’t)

Stream-feed: cognitive advantages for the learner.

- streams are based on what community is producing (discussing, writing) or interested in (papers, images) at any given time

- presented in the order they are posted, each medium is considered a stream.

Stream-Feed Integration (con’t)

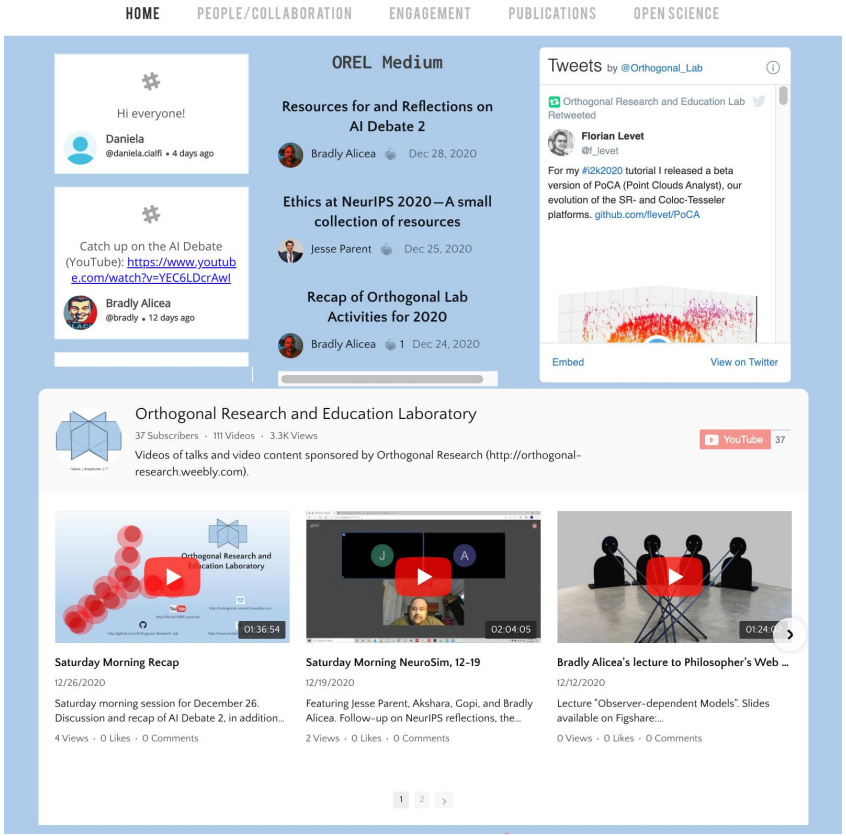
Stream-feed: cognitive advantages for the learner.

- streams are based on what community is producing (discussing, writing) or interested in (papers, images) at any given time
- presented in the order they are posted, each medium is considered a stream.

To achieve a curated feed, the user sorts through the multiple streams manually.

- resulting feed is driven by human curiosity and based on a non-random set of interests

In context, serendipitous themes begin to emerge: interesting posts that might form a coherent topical theme.



Curated Discussion Channels to Support Community (making discussions sustainable)

Slack/Discord Model of Community Discussion

Manage and post links, start convos, lurk and respond to all posts (as a moderator or participant).

- file sharing, chat is indexable and searchable, channel structure and private groups, DMs (as a participant).
- invite people to join (Newsletters, Public Events, other Outreach).
- get people access through link invitation or Launchpass.
- join a channel to become part of the conversation, or tag people to bring their attention to something.

Benefits and Drawbacks

- private wrt the internet, easy to navigate, get feedback and manage content.
- directly answer questions in threaded discussions.
- message avalanche — too many messages per unit attention.
- channel review can tie into your community activities
    - Agile-oriented standup (native to the Slack channel, bot will ask questions once a week and select people will provide an update).
    - formal virtual meeting review (feature Slack channels in a meeting).

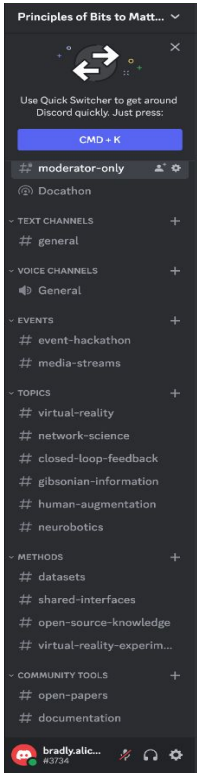
Stay Focus on Slack
https://zapier.com/blog/stay-focused-on-slack/

How to keep Slack channels “optimal”
https://www.pcmag.com/how-to/get-organized-how-to-keep-slack-tidy-scannable-and-not-too-busy

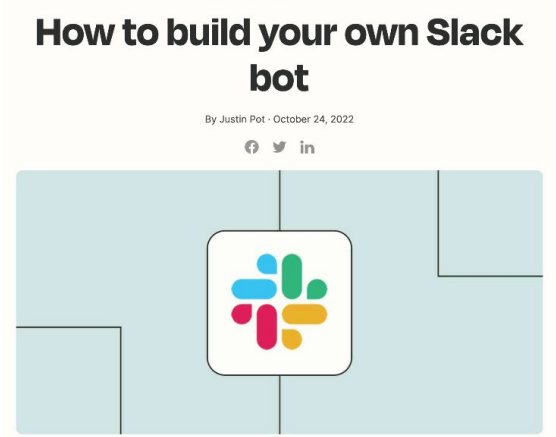
Slack solved the problem of e-mail threads where people lacked control of the conversation flow.

Set up Slack team/Discord server in a way that is manageable.

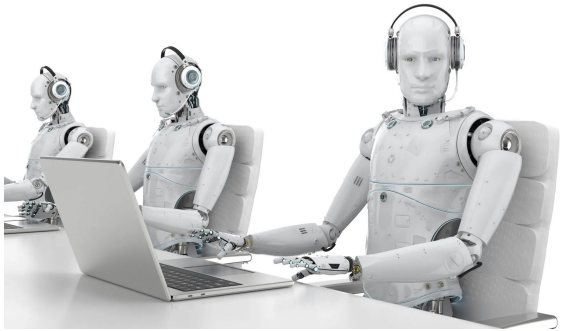
- aggressively *leave* channels you don't need to be in.
- *thread discussions* — and unsubscribe when needed.
- *change* your notification settings.
- declutter and *organize* your sidebar.
- save *actionable* items (and possibly send them to your to-do list).
- schedule away times and *automate*.



Slack/Discord/Mastodon Bots



https://zapier.com/blog/how-to-build-chat-bot/



Programs (Chatbots) that Automate repetitive processes

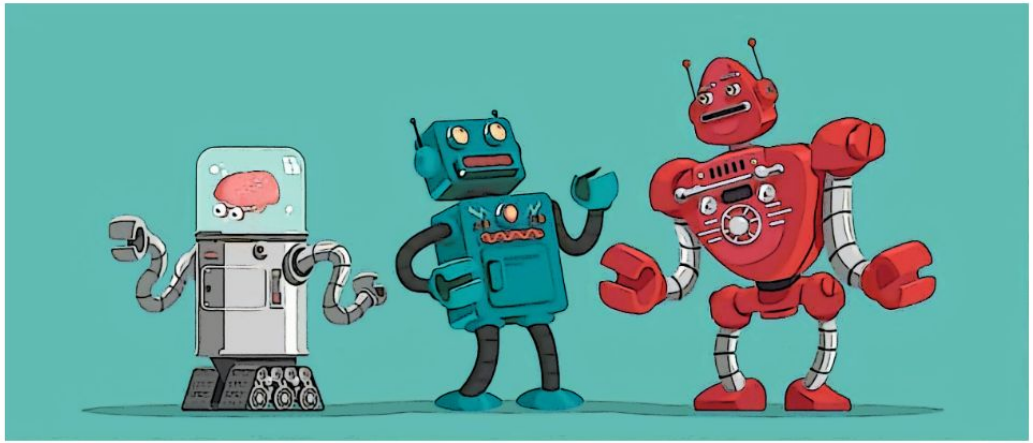
Reply bots: for weekly stand-ups and other types of chat functions.

Notification bots: what's happening outside of your channel or platform?

Slash Command bots: call a bot to do specialized tasks (find gifs, make a video call).

How to Build a Bot and Automate your Everyday Work

Tim Grossmann



Most jobs have repetitive tasks that you can automate, which frees up some of your valuable time. This makes automation a key skill to acquire.

https://www.freecodecamp.org/news/building-bots/

“Likes and engagement are digital currency and you are devaluing them”

Prisoner's dilemma, zero-sum game:

If I don’t bot and you bot, you win. If you don’t bot and I bot, I win. If no one bots, everyone wins.

Since there is no incentive for everyone not to bot, everyone bots, so no one wins.

The Discourse Model of Community Discussion

Examples from INCF Neurostars Discourse

About Neurostars

A question and answer forum for neuroscience researchers, infrastructure providers and software developers.

Our Admins

PradeepGeorge
Admin

Mathew_Abrams

leif Leif Larsson

Our Moderators

effigies Chris Markiewicz

emdupre Elizabeth DuPre

oesteban Oscar Esteban

satra Satrajit Ghosh

malin

rwblair Ross Blair

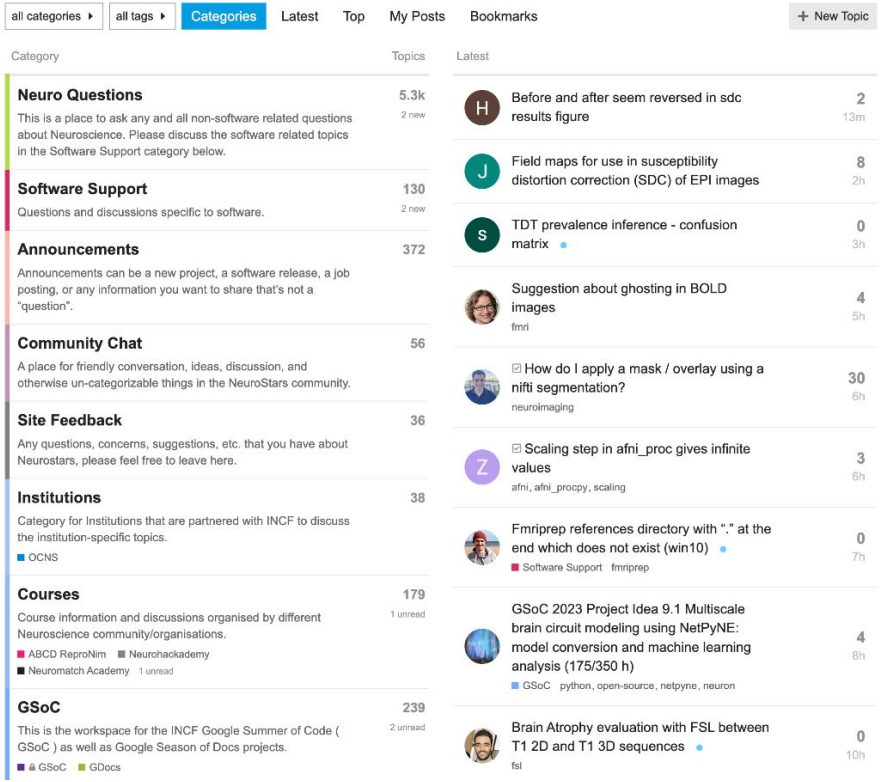
ChrisGorgolewski
Regular

franklin Franklin Feingold

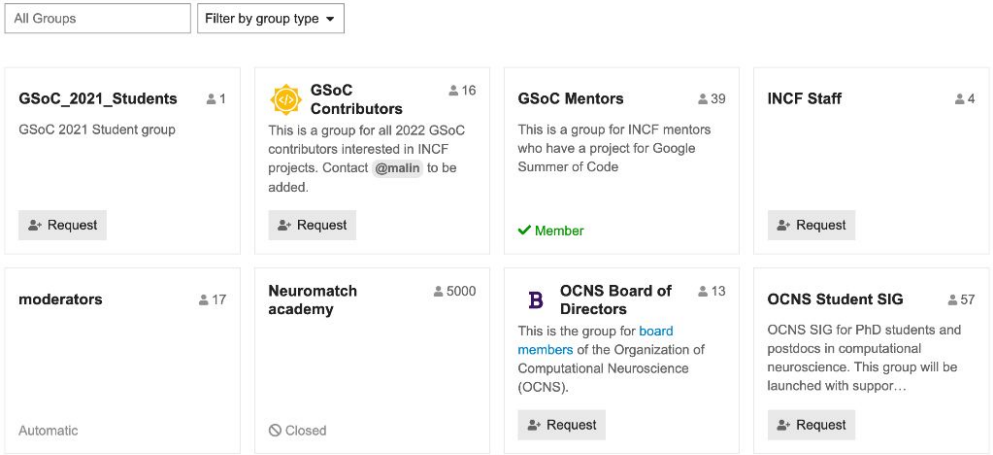
Helena
NMA_admin

Site Statistics

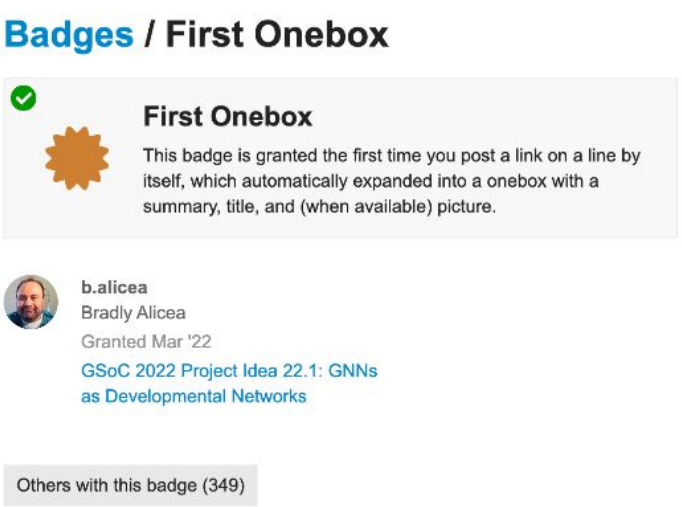
	Last 24 hours	Last 7 days	Last 30 days	All Time
Topics	3	19	139	6.4k
Posts	37	215	941	62.9k
Users	4	28	131	9.5k
Active Users	87	256	498	—
Likes	17	49	151	17.6k



The Discourse Model of Community Discussion



Participants join groups and use badges for basic competencies



Discourse Style of Organizing Community Discussion

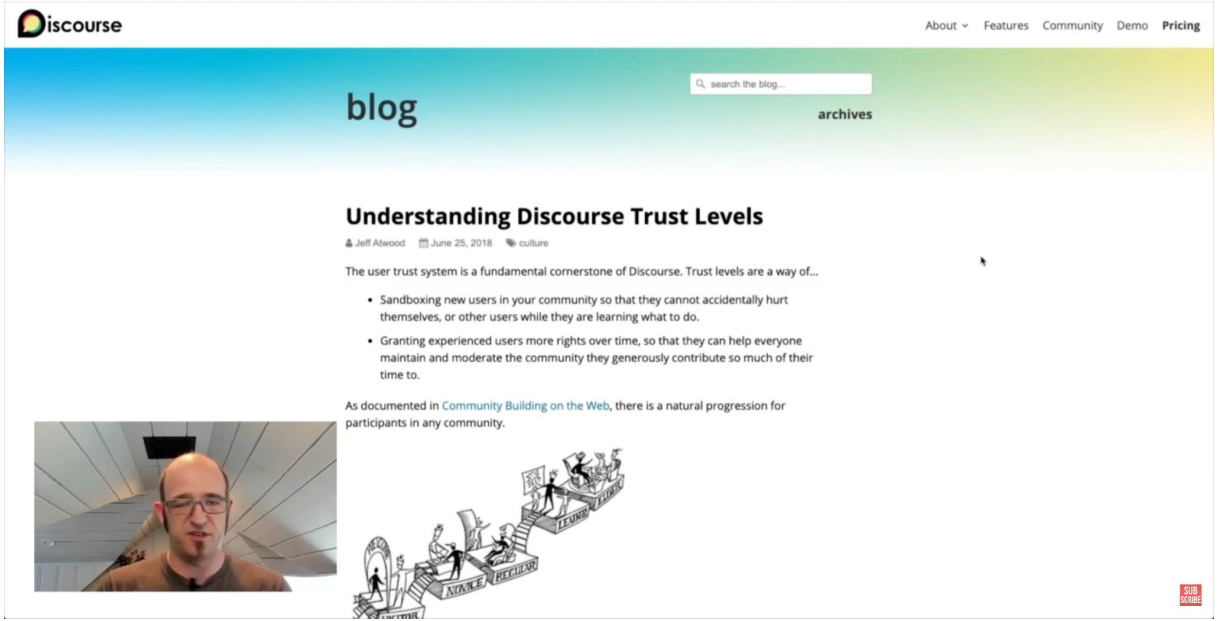
Visibility (interface design matters):

- themed buckets of discussion. Categorization using colored badges. Discovery of similar topics.
- use Markdown for message threads (images and stylized text in discussion).

If you have 1000 users, who is most active and how could you best leverage their efforts?

- most active community members → incentivized for effort.
- less active community members → incentivize to become more active.

Discourse Style of Organizing Community Discussion (con't)



Discourse (user) trust levels

Jono Bacon YouTube: https://www.youtube.com/watch?v=UkxNL--_Q88

Discourse Blog: https://blog.discourse.org/2018/06/understanding-discourse-trust-levels/

Discourse Style of Organizing Community Discussion (con't)

Discourse trust levels (compare with Wikipedia Contribution Type Counts)

Three types and five levels of community member:

- types: casual, regular, and core.
- levels: visitor, novice, regular, leader, elder.

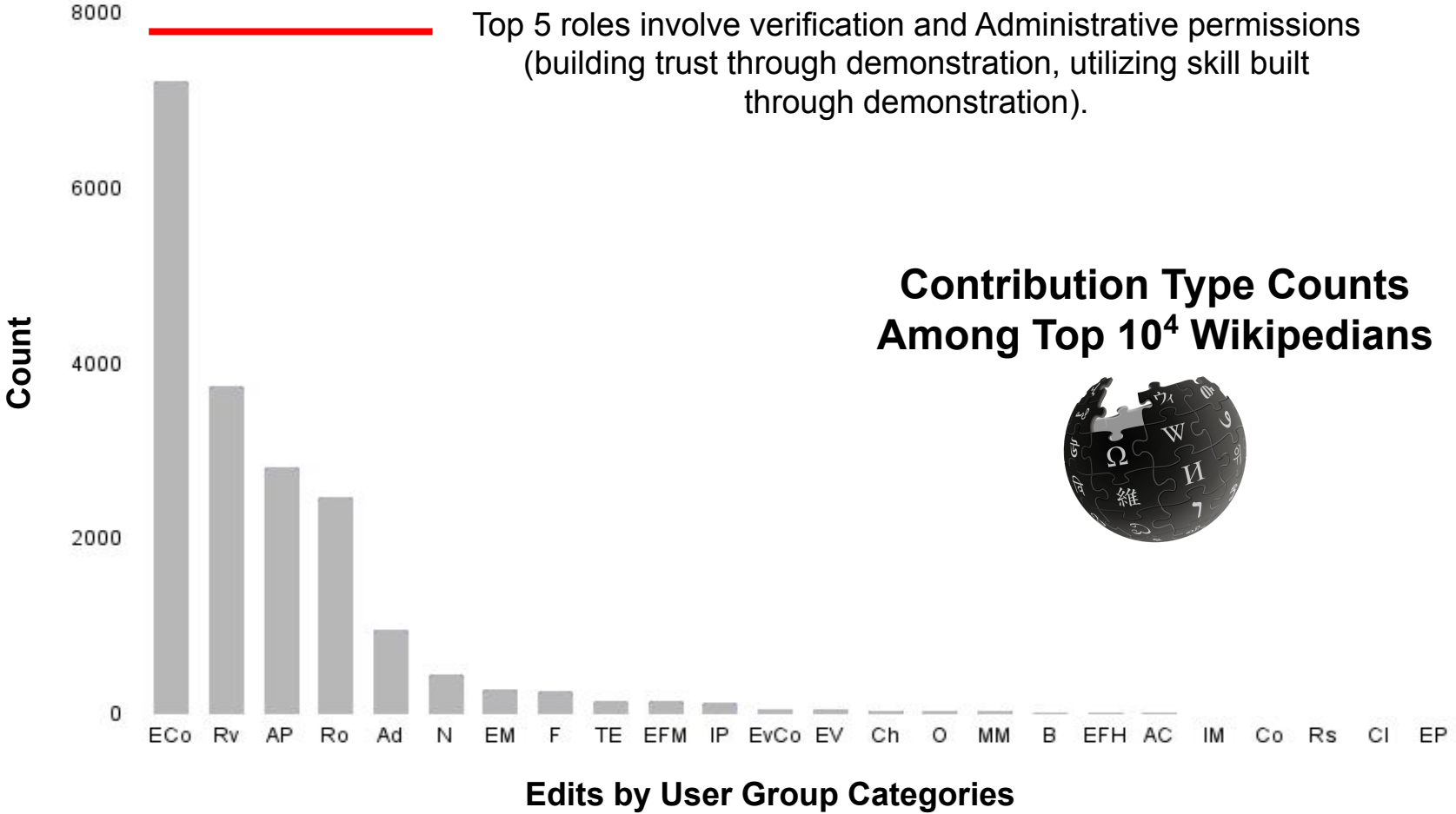
Criterion for becoming a "member":

- badges: visiting on at least 15 days (not sequentially), casting or receiving at least 1 like, replying to at least 3 different topics, entering at least 20 topics, reading at least 100 posts, spend a total of 60 minutes reading posts.

Badges enable community members to do greater things, become more involved (higher on the community commitment curve).

Top 5 roles involve verification and Administrative permissions (building trust through demonstration, utilizing skill built through demonstration).

Contribution Type Counts Among Top 10^4 Wikipedians



Minimum Viable Community: https://www.commsor.com/post/minimum-viable-community

The smallest (but not bare-bones) group of people supporting programs where members will find value.

The smallest group of people needed to come together for a shared purpose.

Minimum Viable Community (con’t)

Tips

- start small and scale as you have more insight into what really works for your members.
- communities are inherently unique in terms of audience, size, and scope.
- take an iterative approach to build community and build over time, but don’t be minimalist to start.

Benefits

- resources don’t need to be stretched.
- a better understanding of customer needs.
- get feedback and scale faster.