

Open Project Management

from an “open” perspective

UNIT 2

Instructor: Dr. Bradly Alicea

<http://bradly-alicea.weebly.com>



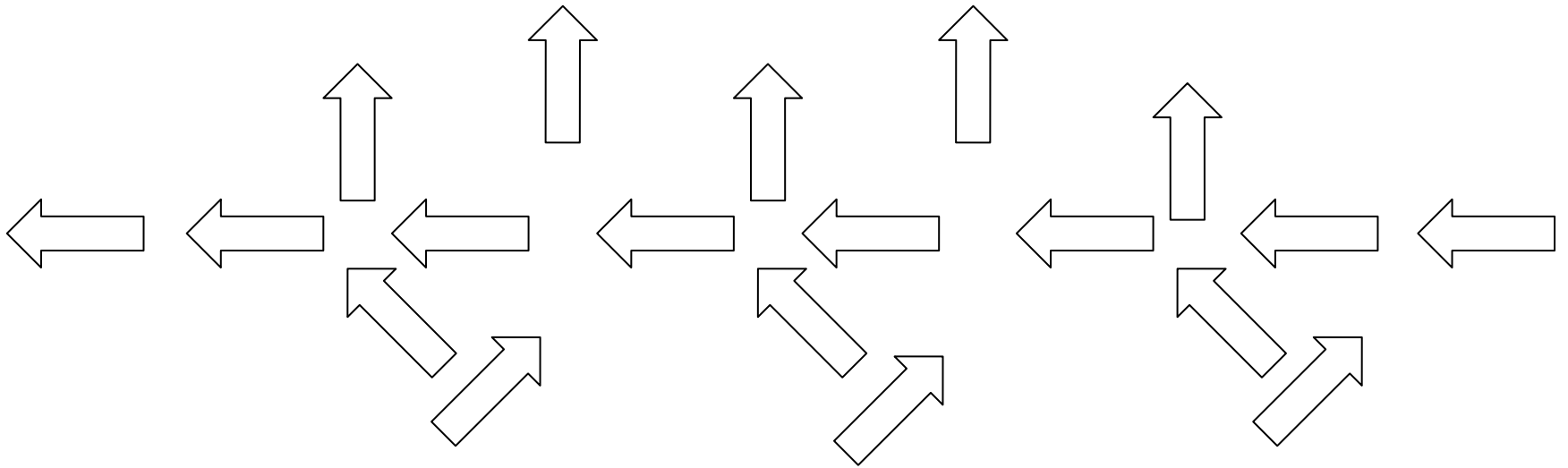
IS 340: Spring 2023

All content



Open Project Management

Welcome Back!



Open Project Management

Welcome Back!



Gamification and Community-building

Top 5 Open-source Tools in 2021

<https://raccoongang.com/blog/top-5-open-source-gamification-tools-2021/>

Gamification

<https://www.valamis.com/hub/gamification>



Gamification focuses on:

- Goals, Status, Building Fellowship, Education, Rewards

Successful gamification:

- taps into intrinsic motivation (become more skilled at the job).
- offers an extrinsic motivation (rewards, points and badges).

**A last word about metrics and facilitating
cooperation, social capital, and working open**

Agent-based Model (Iterative Prisoner's Dilemma) for Sharing Open Data

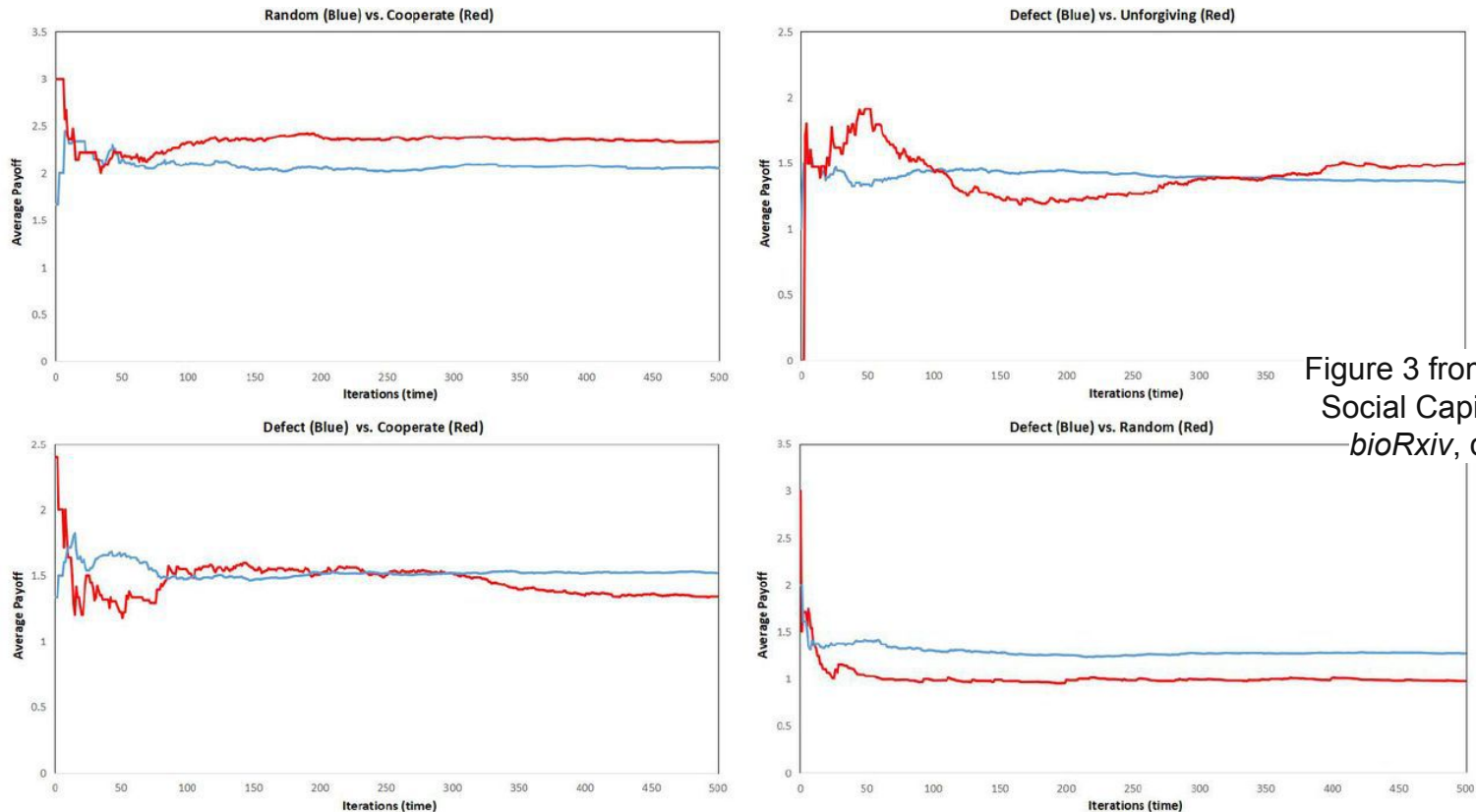


Figure 3 from “Data Reuse and the Social Capital of Open Science”,
bioRxiv, doi:10.1101/093518

Technical Debt and Reducing Friction

Communities and Technical Debt

Implied cost of additional rework caused by choosing an easy (limited) solution now instead of using a better approach that would take longer.

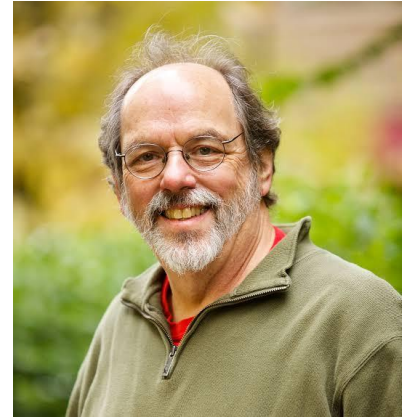
- when ignored, debt accumulates over time (negative interest), becoming harder to pay off.

<https://www.techopedia.com/definition/27913/technical-debt>

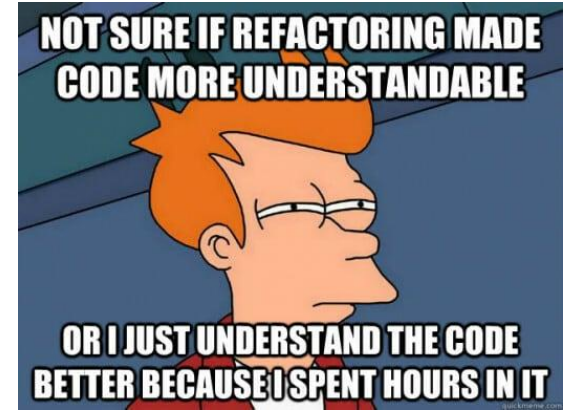
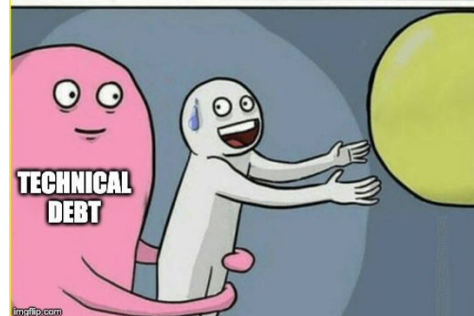
Coined by Wiki creator Ward Cunningham. Origins in extreme programming and refactoring. Also a common topic in Scrum

<https://www.scrum.org/resources/blog/technical-debt-scrum-who-responsible>

Who is responsible for dealing with issues that lead to technical debt, and who is responsible for paying it down?



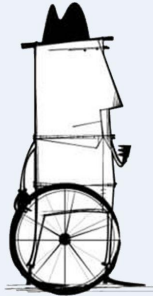
Communities and Technical Debt (con't)



WHAT IS TECHNICAL DEBT?

Technical debt can be defined as whenever software is developed in a partial way that later produces errors and artifacts that needs to be addressed with a stable process and features. In other words, it's the result of prioritizing speedy delivery over perfect coding.

ERRR...



*CAN'T STOP.
TOO BUSY!!*

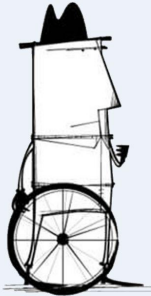


<https://www.chapter247.com/blog/with-great-technology-comes-technical-debt/>

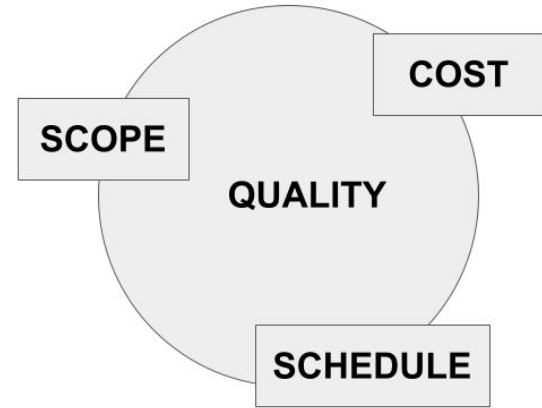
WHAT IS TECHNICAL DEBT?

Technical debt can be defined as whenever software is developed in a partial way that later produces errors and artifacts that needs to be addressed with a stable process and features. In other words, it's the result of prioritizing speedy delivery over perfect coding.

ERRR...

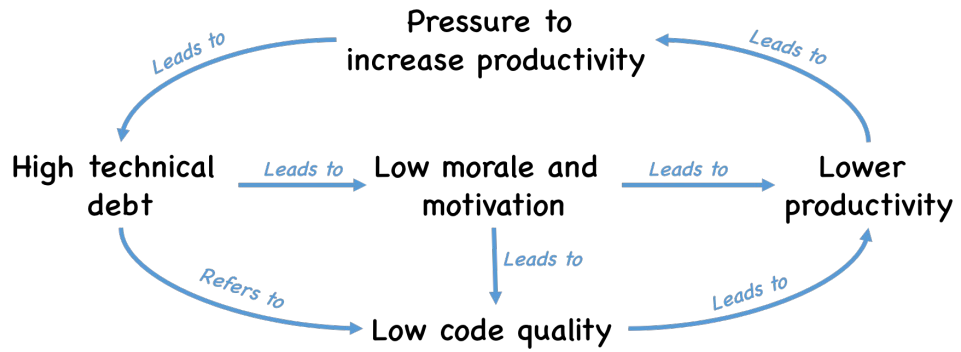


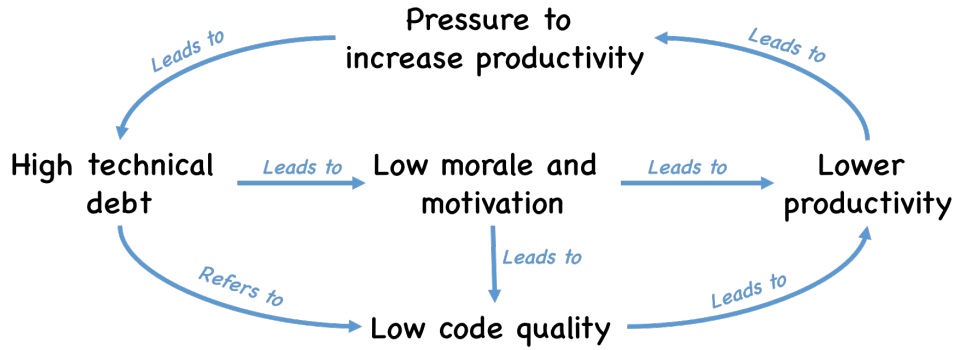
*CAN'T STOP.
TOO BUSY!!*



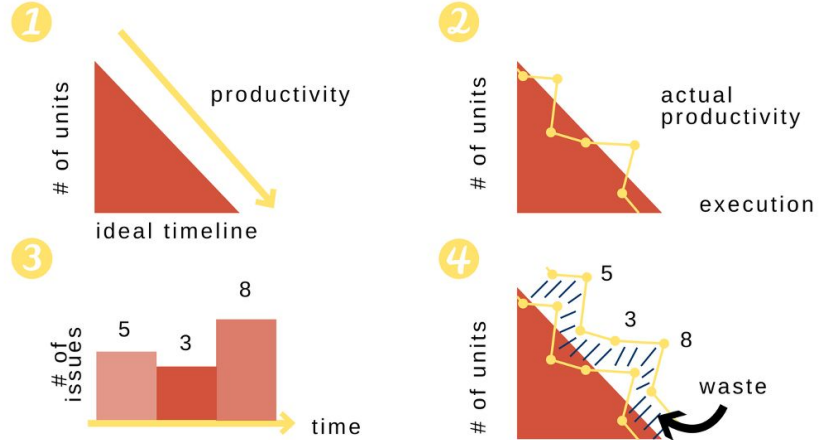
Debt can be incurred if it can be managed:

- need to implement quickly.
- lack of resources to work in an optimal fashion.





MAKE WASTE VISIBLE



Making Technical Debt Visible

<https://www.scrum.org/resources/blog/making-tech-debt-visible>

The screenshot shows a GitHub repository page for 'mrdourke/food-not-food'. The repository is public and has 2 unwatched items, 13 stars, and 1 fork. The main content area displays a file list for the 'main' branch, with 1 commit by 'a8e353e' 4 days ago. The files listed include .gitignore, LICENSE, README.md, data_exploration.ipynb, data_splitting.ipynb, food_not_food_model_v3.tflite, hotdog-not-hotdog.jpeg, imagenet1000_clsidx_to_labels.txt, index.html, model_building.ipynb, and script.js. A video player is overlaid on the bottom half of the page, showing a video titled 'Cleaning up foodnotfood.app GitHub Repo | Python coding | Data cleaning (live)' with a progress bar at 3:46 / 3:31:34. The video player also shows a description: 'See notes on Notion: https://www.notion.so/mrdourke/November-6-100k-Livestream-Celebration-a8e0836c7a9490f94ada891e606d8e'.

Cleaning up foodnotfood.app GitHub Repo | Python coding | Data cleaning (live)

D Daniel Bourke arXiv
1.62K subscribers [Subscribe](#)

[👍 4](#)
[🗨️](#)
[🔗 Share](#)
[⬇️ Download](#)
[✂️ Clip](#)
[⊕ Save](#)
[⋮](#)

152 views 1 year ago
In a recent livestream I built an app called foodnotfood, a simple app to detect whether an image is of food or not.


Cleaning up the repo for the “Food Not Food” App
<https://www.youtube.com/watch?v=-4yq486ZaPU>

Reducing Community Friction From the “Start Here”



Asking Your Community for Help


Posts and replies ← Back

 **Anne Urai**
@anneurai@neuromatch.social

A skill we rarely teach, but is imo crucial, is how to effectively ask for help from online software communities.

Any best practices guides to get started, and go beyond 'this is some code, can you fix it?'

Feb 15, 2023, 05:59 · 7 · 6

 **Florian Idelberger**
@fl0_id@mastodon.social

@anneurai provide an exact description of what you did. Any error messages, inline or if too large in an external file. Ideally als specify what you already tried, and try to search at least before asking, on a search engine or/and the community. If they have a specific process or template for questions, use that. (Oh and that might often say to only report/ask about the newest version but m that depends)

 **JD Long** ✓
@Cmastication@mastodon.social

@anneurai strong agree. We added a recipe in R Cookbook 2E on this very thing

rc2e.com/gettingstarted#recipe...

 1 Getting Started and Getting Help | R Cookbook, 2nd Edition
rc2e.com

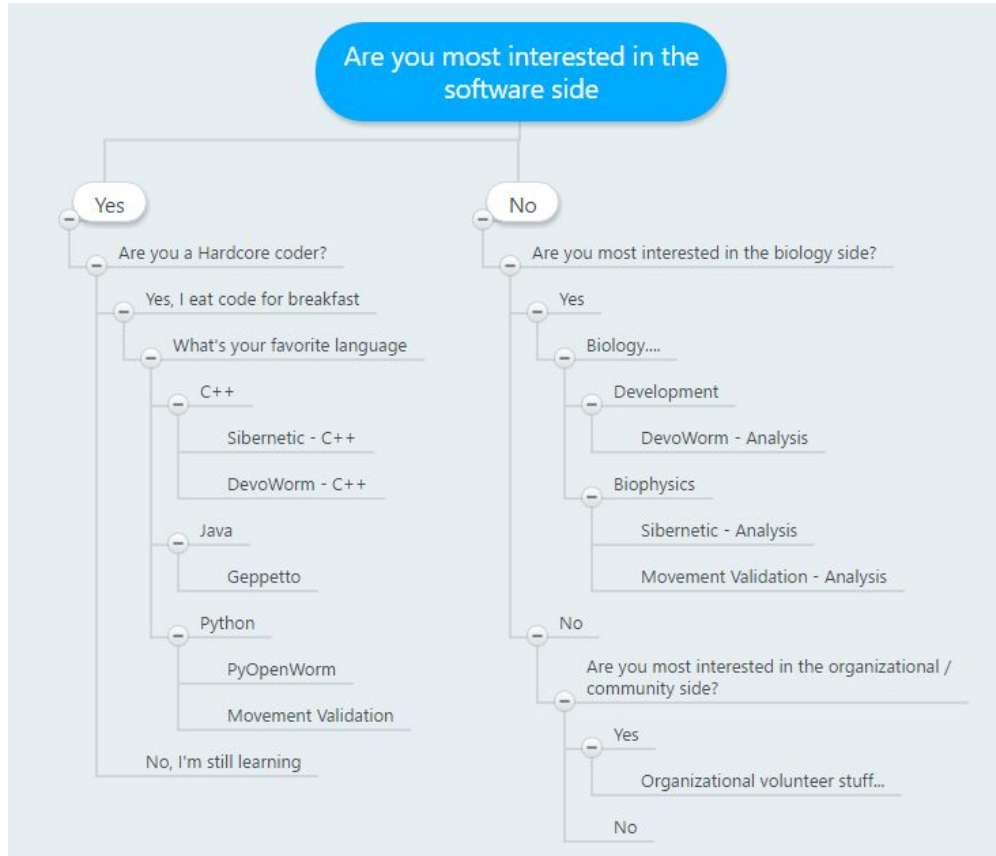
 **Dan Goodman**
@neuralreckoning

@anneurai the most important thing is to (a) explain what you've already tried and didn't work to save time and demonstrate that you have tried to solve the problem on your own, and (b) provide a minimal example that demonstrates the problem, because nobody is going to read hundreds of lines of code for you but they might read 10 lines.

Submitting Questions to Stack Overflow or Elsewhere in the Community (R Cookbook)

<https://rc2e.com/gettingstarted#recipe-id269>

Guidance Tree



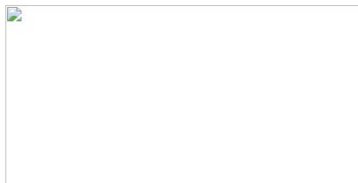
Community Committee, circa 2016
Stephen Larson's sketch

Guidance Tree to funnel potential contributors into a specific place

Guidance Tree (in practice)

[Community Navigator](#) [Developers](#) [Onboarding](#) [Research](#) [Education](#)

Welcome to the Rokwire Community Guidance Tree!



Would you like to make a coding (programming) contribution, or a building (documentation, maintenance, design, or discussion) contribution?

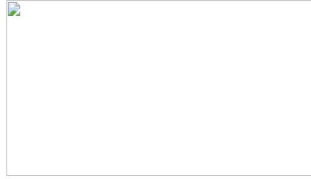
Coding

Building

Beta Version. Suggestions? [Contact us](#). Welcome to the root page!



All content licensed under Creative Commons license [CC-BY 4.0](#).



Are you interested in platform adoption, or making an open source coding contribution

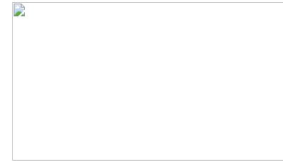
Platform Adoption

Open Source Coding

Beta Version. Suggestions? [Contact us](#). Want to start over? Back to the [Root Page](#).



All content licensed under Creative Commons license [CC-BY 4.0](#).



What kind of building contribution would you like to make: a technical review contribution (documentation, quality assurance), or a ideation-related (discussion, design) contribution

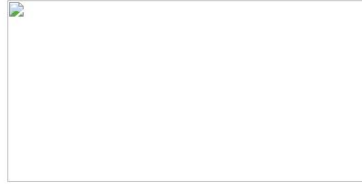
Technical Review

Ideation

Beta Version. Suggestions? [Contact us](#). Want to start over? Back to the [Root Page](#).



All content licensed under Creative Commons license [CC-BY 4.0](#).



Have you consulted our Wiki, Docs, Tutorials, and Rokwire API resources?

Yes

Developer Resources

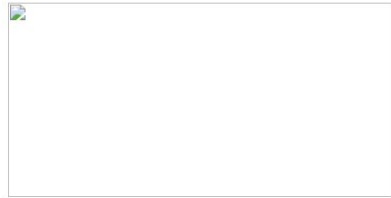
Beta Version. Suggestions? [Contact us](#).



All content licensed under Creative Commons license [CC-BY 4.0](#).

Github Repository

Developers Tab



Would you like to be involved in creating demos/tutorials, or contributing to and editing documents?

[Demos/Tutorials](#)

[Wiki Documents](#)

Beta Version. Suggestions? [Contact us](#). Want to start over? Back to the [Root Page](#).



All content licensed under Creative Commons license [CC-BY 4.0](#).

Community Building: Introduction

An introduction* to the tools, systems, and processes of open-source community building.

* some of these topics in this unit will be revisited in more detail in later units.

Community Building: Introduction

An introduction* to the tools, systems, and processes of open-source community building.

* some of these topics in this unit will be revisited in more detail in later units.

A community is a group of collaborators that can be integrated with project development.

Community is generally involved in building, supporting, and promoting the project.

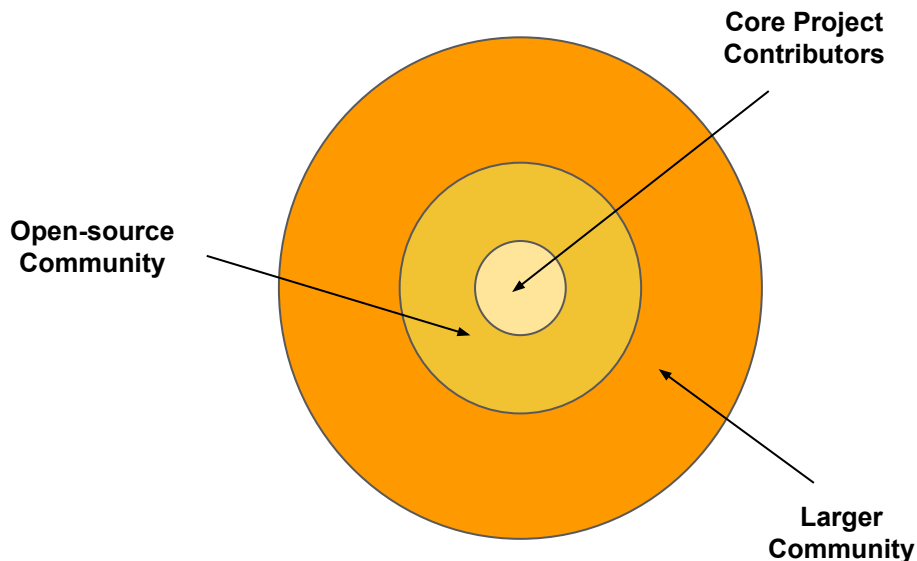
Community Building: Introduction

An introduction* to the tools, systems, and processes of open-source community building.

* some of these topics in this unit will be revisited in more detail in later units.

A community is a group of collaborators that can be integrated with project development.

Community is generally involved in building, supporting, and promoting the project.



Community Building: Introduction

Build Open Project, then Community

- History of Project Management.
- Why Community Building?
 - bug reports and community safety examples.

Community Building: Introduction

Build Open Project, then Community

- History of Project Management.
- Why Community Building?
 - bug reports and community safety examples.

Paradox of Open and the role of a Community

- commons as an ideal, communities of practice, open does not equal success in ideals.

Six Principles of Building a Community

- comparison, engagement, evangelism, state, behavior.

Community Building: Introduction

Building a Community (con't)

- tradeoffs, constraints, and project scope.
- points of failure and institutional knowledge (avoid top-down centralization).

Community Building: Introduction

Building a Community (con't)

- tradeoffs, constraints, and project scope.
- points of failure and institutional knowledge (avoid top-down centralization).

Gaining Momentum and Buy-In

- social capital (altruism, innovation, trust)
- Localization example (foreign-language translations).

Community Building: Introduction

Building a Community (con't)

- tradeoffs, constraints, and project scope.
- points of failure and institutional knowledge (avoid top-down centralization).

Gaining Momentum and Buy-In

- social capital (altruism, innovation, trust)
- Localization example (foreign-language translations).

Automation

- onboarding, cultivating leadership, community patterns through data.
- Discord community example (values and go-to person).

Community Building: Introduction

Big Picture and Community Planning

- Community Commitment Curve.
- Master Plan (incentives, sustainability, catalyzation, communication/innovation, and embedding).
- community/project synergy.

Community Building: Introduction

Big Picture and Community Planning

- Community Commitment Curve.
- Master Plan (incentives, sustainability, catalyzation, communication/innovation, and embedding).
- community/project synergy.

Measures and Metrics

- Bulk Github activity, dashboards, fork trees.
- putting contribution skills into practice – dashboard.
- Agent-based models (simulating potential outcomes).

Community Building: Introduction

Big Picture and Community Planning (con't)

- One-many interactions: where contributor fits into a working group.
- networks, network science applications.

Community Building: Introduction

Big Picture and Community Planning (con't)

- One-many interactions: where contributor fits into a working group.
- networks, network science applications.

Technical Debt

- What is it? Constraints on project and community development.
- How to minimize (reducing friction), evaluate, and measure.
 - cleaning repos, asking for help from community.
 - guidance tree.