

Governance & Decision Making






Building effective structures for open innovation

Lars Schöbitz

Global Health Engineering, ETH Zurich

July 15, 2025

Session Roadmap (70 min)

-  Introduction & Context (~10 min)
-  Governance Models (~15 min) - Including
-  Activity 1 (5 min) - Project Assessment
-  Roles & Code of Conduct (~10 min)
-  Meeting Management (~15 min) - 10 rules for meetings

Introduction

Attribution

These slides are reproduced from github.com/codebender “Copyright (c) Greg Wilson”. The slides are licensed under <http://creativecommons.org/licenses/by/4.0/>

Starting point

A 10X engineer is one who can bring together engineers and emerge with a shared understanding and consensus of the problem being solved and how to be done.

— [Lorin Hochstein](#)

In this house we call them “project manager”

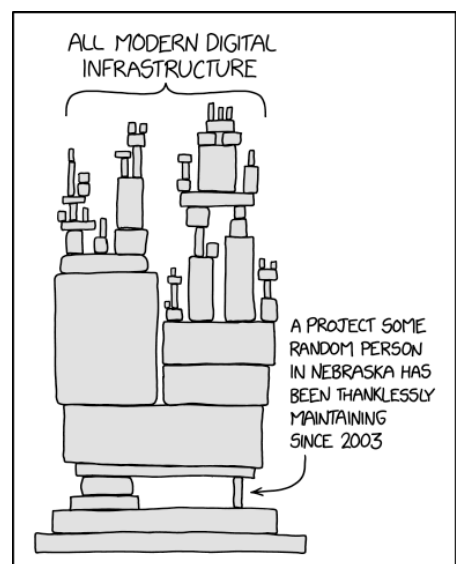
— [Greg Wilson](#)

</> Research software (because we create)

- Software is created and run to answer a question
- Papers, theses, and other reports are the products
- The software is “just” a tool
- Shades into projects that produce software researchers to use

You

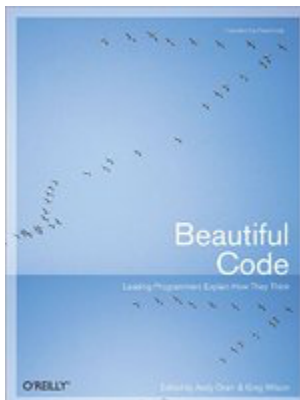
- Degree(s) in some research domain
- Little or no formal training in software development
- In a team of 1–12 people
- Timelines of weeks to months



XKCD Dependency cartoon



- Don't remember much statistics...
- ...but I have programmed a bit in R,
- ...I've managed a few software projects,
- ...and I've been lucky enough to hang out w people



Where we've been, where we're

Then	Now	Ne
Dropbox	Git repository	Br
“Just do it”	Slack / mailing list	Ma
Interactive analysis	A big pile of scripts	Bu
Word / Google Docs	Notebooks / LaTeX	Sit
“It doesn’t crash”	“Are there any NAs?”	As
“Um, hi?”	README + LICENSE	CO Co

Acknowledgments

- Original workshop: Daniel Standage and Greg Wilson
- *Producing Open Source Software*: Karl Fogel
- *Research Software Engineering with Python*:
Kate Hertweck, Luke Johnston, Joel Ostblom,
Charlotte Wickham, and Greg Wilson



Cover of 'Research Software Engineering with Python'

Activity: Project Assessment

In our shared Google Doc, Section: “Govern

Reflect on your current project:

1. Who uses your software / your work?
2. How do they find it?
3. Who decides what will happen next?
4. How is that communicated?
5. Who can make what kinds of changes?
6. What happens automatically?
7. How are newcomers brought on board?



Governance

What problems are we trying

- Low productivity
 - “Oh no: not another meeting...”
- Opaque decision making
 - “Did we decide that?”
 - “It’s not what you know, it’s who you know”

👍 The good news

- You don't have to invent this yourself
- Learn from others who've solved these problems
- <https://www.askamanager.org/> - Great resource for advice

Example insights: - "Consensus doesn't mean it means everyone can live with the decision" decisions immediately, or they didn't happen"



ASK A MANAGER
and if you don't, I'll tell you anyway

Governance models

- <https://communityrule.info/> describes [lots of](#)
- Benevolent dictator (often the project founder)
 - Common in young projects
 - Brittle (founder can move on)
 - Usually leads to emergence of unofficial leaders
- Elected representation
 - Explicit rules for suffrage
- Consensus-based
 - If most people agree on most things most

“Hero” programmers

- Brooks advocated a chief programmer model ([Brooks 1995](#))
 - Disparaged since then
- But 80% of projects on GitHub are hero projects ([al. 2019](#))
 - 5% or less of people responsible for 95% interactions
 - “Heroes” commit far fewer bugs than others
- Despite terminology, not a bad model for research
 - ([Petre and Wilson 2014](#)) found that people without domain knowledge couldn't review scientific code

Martha's Rules

- Anyone can put forward a proposal by filing before a scheduled meeting
 - One-line summary
 - Background information
 - Concrete proposal
 - Pros and cons
 - Alternatives
- At most two pages
 - Preferably shorter

Establishing a quorum

- A quorum is established if half or more of v present
- Which means there must be:
 - A rule about how to become a member
 - A rule about when and how someone sto
- The meeting may not discuss or vote on a p sponsor (or their delegate) is present.

Presenting a proposal

- The sponsor has 5 minutes to present the proposal
- Members cast a sense vote: support, neutral, or oppose
 - If everyone supports or is neutral, go immediately to the final vote with no further discussion
 - If a majority is opposed or neutral, send the proposal back to the sponsor for further work
- If a minority of members oppose, set a time for a moderated discussion
- Then call a final binary vote in which everyone must either support or oppose (no neutral votes allowed)
- If a majority support, the proposal is accepted
 - Otherwise, it is returned to the sponsor for further work



Activity: Decision Making (5

In our shared Google Doc, Section: “Decision

1. What decisions has your project made rece
2. Who made them?
3. Where are they recorded?

☰ Roles and responsibilities

People & Their Roles

Person	Roles
ghopper	admin
kjohnson	admin, commit
aturing	commit
bwk	commit

Roles &

Role
admin
commit

Benefits:

- Gives you a list of what actually needs to be done
 - It's always longer than you first expect
- Tells everyone who to go to for what
 - Particularly when their first choice is on leave
- Helps with succession planning
 - "We don't have anyone who does that any more"

Activity: Roles & Responsibilities (15 minutes)

In our shared Google Doc, Section: “Roles & Responsibilities”

Column A: Tasks List things people do to keep your project going	Column B: Contributors List names of contributors
--	---

Then: Draw connections - what patterns emerge?

Code of Conduct

- Diverse communities need explicit norms
 - “I didn’t realize someone would find that
 - Which is sometimes used dishonestly...
- Adopt a [Code of Conduct](#) (e.g., [Python](#)’s)
 - Its mere existence is a strong signal about the community you are
- Don’t write one yourself
 - You won’t think of all the edge cases
 - Using someone else’s makes misunderstanding
- Use the [Contributor Covenant](#)

Code of Conduct (continued)

- A Code of Conduct is only useful if it is enforced
 - And if there is a clear reporting mechanism, members trust
- Be explicit about enforcement mechanisms
- Designate an independent third party to handle
 - Trade off with another project lead
 - *Do the background reading* ([Aurora and C](#))

Meetings

- On par with interruptions for “things people have less of”
- Unlike interruptions, can be done well
- As with governance, having rules is the first toward efficiency

❓ 1. Does there actually need to be a meeting?

- To *inform*? Only if you are expecting questions
- To *consult*? Only if people get a vote
 - Otherwise it's just informing with pretensions
- To *discuss* and *make decisions*? Yes
 - But only in small groups
 - Or with well-defined procedural rules
- To *brainstorm* or *collaborate*?
 - That's a very different kind of meeting

☰ 2. Create an agenda

- If you don't care enough to make a list, you're not going to have a good meeting
- Include timings
- Prioritize
- Plan to end early
 - “The most fundamental unit of time is the hour”



3. Have clear rules for making

- “The Tyranny of Structurelessness” ([Freema](#)
- If you need Robert’s Rules, you need training

4. Put someone in charge

- The moderator should *not* do most of the talking
 - Any more than the conductor plays most of the music
- Call on specific people in order
- Allow them one point at a time
- Keep a backlog

5. Require politeness

- All the other rules are special cases of this.
- No technology during in-person meeting
 - Except for assistive technology or family
 - “Please put your devices in politeness mode”
- No interruptions
 - Except by moderator
- No rambling
- You *do* have a Code of Conduct, right?

6. Record minutes

- So people who weren't there know what happened
- So people who were there agree what happened
- So people can be held accountable at later

7. Manage “that guy”



Dealing with Disruptive Behaviors



Talkative



Arrogant



Complaining



Eager



Argumentative



Shy



Diverting



Indecisive



Apathetic



Dominating

- The mode
- Three stic
- Interruption
- <https://coas>

8. Be an active participant

- Decline invitations
 - *If you agree to abide by what the meeting*
- Read the agenda and material before the meeting
- Take your own notes
- Use participants' names
- Pause before speaking
- Put down your hand

9. Life online

- No mixed-mode meetings
 - All in person or all online
- Do not record the meeting without willing consent
- Review meeting protocol at the start if necessary
- Take minutes in a shared document
- Raise hands digitally
 - `/hand` in the chat is good
 - `/hand another budget item` is better

Q 10. Seek truth, not victory

- No social dominance displays
 - “Well actually...”
- Don’t raise points you don’t actually believe
 - The devil doesn’t need more advocates
- Don’t make excuses for your questions or o
 - “This is probably stupid, but...”

Recommended reading

- ([Brookfield and Preskill 2016](#)) has a lot of good managing discussion
- [Producing Open Source Software](#) by Karl Fogel
- [The Tyranny of Structurelessness](#) by Jo Freeman

Key Takeaways

- Governance is not bureaucracy - it's clarity
- Start with Martha's Rules for decision making
- Document roles and responsibilities explicitly
- Make meetings count with clear structure
- Code of Conduct is non-negotiable



Activity: Governance Structure

In our shared Google Doc, Section: “Governance Structure”

1. Who gets a vote in your group?
2. How are new people added to that pool?
3. When and how do people lose their votes?
4. Where do people find out what has been decided?
5. How can non-voters raise issues?

References

- Aurora, Valerie, and Mary Gardiner. 2019. *How to Respond to C*
Version 1.1. Frame Shift Consulting LLC.
- Brookfield, Stephen D., and Stephen Preskill. 2016. *The Discuss*
Get People Talking. Jossey-Bass.
- Brooks, Frederick P., Jr. 1995. *The Mythical Man-Month: Essays*
Anniversary edition. Addison-Wesley Professional.
- Freeman, Jo. 1972. “The Tyranny of Structurelessness.” *The Sec*
- Majumder, Suvodeep, Joymallya Chakraborty, Amritanshu Agra
“Why Software Projects Need Heroes: Lessons Learned from
abs/1904.09954.
- Petre, Marian, and Greg Wilson. 2014. “Code Review for and by
Workshop on Sustainable Software for Science: Practice and
[arXiv:1407.5648](https://arxiv.org/abs/1407.5648).