



# Maintainers III

## MONDAY

### Maintaining FITS in Two Careers (Panel)

(Bernadette Boscoe, Michael Scroggins, Mehdi Medjaoui (moderator), Don Goodman (moderator))

FITS developed in the 70s. Interoperability was an issue, because of vendor lock-in.

Maintenance was always, and continues to be, seen as a burden.

- Good maintenance always hindered by management, not by technical ability.

FITS

- Always has image and audio
- always backwards compatible
- Managed by groups of astronomers.
  - Used to be academic institutions, now volunteer associations
  - group of about 20-30.
  - stable core of people, but who are in precarious employment positions: not tenured, grad students, post-docs doing coding as part of their work in the astronomy field, which contributes to their position being precarious
  - a key concern is maintaining disciplinary knowledge, generation to generation, with such a small group
  - Use by non-astronomers (particularly the Vatican) creates some anxiety that the control will shift away from astronomers to the ISO
- Designed with the intention to preserve knowledge for the future

### Invisible Work in Highly-Visible Roles? Examining Burnout in Free/Open-Source Software Maintainers (Presentation)

(Stuart Geiger, Mehdi Medjaoui (moderator), Don Goodman (moderator))

stuart@stuartgeiger.com | don't tweet me plz

maintainers end up doing legal, social media, community management, etc, etc

burnout - for both famous and not maintainers

\* it's too much -- bus factor - barrage of requests - unable to take a break - social exhaustion - -

\* it's not fun -- lack of recognition, conflict, not the creative work



## Software Maintenance Collaborative Notes Document

**themaintainers.org**

\* it's not worth it - personal issues, project doesn't help job, project life balance (prof/project overlap unlike volunteering at a soup kitchen)

os notifications - vulnerability - increase issue count

because so much work is tracked on public platforms it's easier to forget the other work

taguette.org -- data analysis

political issues - unexpected burden - eg chef/ICE, or even just licensing issues

not linked to burnout but a form of stress, emotional drain.

you can now say you are on vacation on github

maybe instead of making the invisible labour visible, we should make the visible labour less visible

github - allcontributors

<https://allcontributors.org/>

what does celebrity maintenance (the Kardasification of maintenance) mean going forward?

- some are self aware & raise others up

- OS using 'maintainer' as a catch all term, capturing things like commit rights, or whatever

- maintainership with a notion of care and responsibilities

- relation to academic model, eg editor of journal.

- being able to be the person who speaks at a conference for the project

- if your users are a mix of academics and industry - is publishing a thing? eg proj jupyter - project can act as lead author - consortium model

- instruments get cited in astronomy

### **Unconference Topic Planning**

(Mehdi Medjaoui (moderator), Don Goodman (moderator))

Text

### **Unconference 1 Topic Seed: A Valuable Flaw: Bug Bounty Programs, Software Maintenance, and Infrastructure Labor (Panel)**

(Ryan Ellis, Yuan Stevens)

@bug\_research | [r.ellis@northeastern.edu](mailto:r.ellis@northeastern.edu)

[@ystvns](mailto:@ystvns) | [yuanstevens@protonmail.com](mailto:yuanstevens@protonmail.com)

[Breakout 1 -- Researching Precarity](#)

*Notes. All participants have said they are OK with these notes being public.*

**themaintainers.org**



There is a big culture of volunteerism in F/OSS and elsewhere, is this exploitation? How do you respect people's views while raising issues? Many people feel empowered by this work.

If you're coming from a political economy perspective, analysis is inherently political -- you are showing where capital flows. But in cultural studies, there is a lot on the different kinds of cultural capital, feelings of empowerment, ways of participating, that aren't just economic.

You can get a lot out of doing F/OSS work that benefits companies, you learn how modern software engineering works, you learn the lingo, you make connections -- like an unpaid internship.

People do see it as like an unpaid internship, it is an investment in your future.

There used to be an ethos of open source as a public good, you do it b/c you care about the commons. Now it is about investment in your future career. Although these two co-exist and cause some tensions.

In bug bounties, people put it on their resume, their websites. They build a brand around it.

In Canada, there is a strong law and culture around harm reduction. In security, there is a discussion around making this parallel to decriminalizing sex work. Is this a good metaphor?

There is a big complicating factor with the big tech companies

Similar issue around collective labor -- bug bounties started when sec people said "no more free bugs". But security is a very individualistic field.

Q to bug bounty researchers: How do practitioners respond when you tell them your findings? A: We haven't yet, but we hear the same kinds of complaints. Like issues around companies accepting bugs as valid, legitimate, important, or are told they already know about them.

Q: How does the bug bounty community overlap w/ selling of exploits.

A: Yes, but we only ask about legal things.

The relationship of security to SWE is weird and liminal.

Celebrity is also a big issue in both bug bounties and F/OSS more broadly.

Newcomers are told that they could one day be the big celebrities. They also don't want to join existing projects, they want to start new projects where they can be the BDFL.



**themaintainers.org**

Non-English language: this is also a huge issue. In bug bounty, you do have to negotiate and advocate. Some people join F/OSS as a way to learn English, participate in the global tech economy from a developing nation. Although very few “make it” to a paid position.

How do you write this up in ways that doesn’t victimize people. Focusing on structural issues, talking about how systems disadvantage people. Could be better.

Who is precarious? There can be feelings of relative inequality in tech when salaries can go up to \$500k

The people who are the victims aren’t the bug bounty participants, but the people who have to use the software that is not properly securitized.

### **Unconference 2 Topic Seed: Versions of Versions (Presentation)**

(Mahmoud Hashemi, Mehdi Medjaoui (moderator), Don Goodman (moderator))

Text

Notes:

<https://semver.org/>

<https://calver.org/>

<https://0ver.org>

[https://sedimental.org/designing\\_a\\_version.html](https://sedimental.org/designing_a_version.html)

<https://dafoster.net/articles/2015/03/14/semantic-versioning-vs-romantic-versioning/>

<https://sentimentalversioning.org>

#### Breakout 2 -- The Maintainers’ Duty/Obligation; or, “Good Maintenance”

Obligations that users/stakeholders feel maintainers have to them:

- Responding to pull requests (in a reasonable amount of time)
- Respond to issues / bugs
- Obligations to your team members or others who will have to interact with the code
  - “no broken windows” -- you shouldn’t leave the code messier than you found it.
  - This happens more in companies.
    - In those companies, is there onboarding for this?
- Documentation
  - How do you know what “good docs” are?
  - There are many different kinds of docs
  - Building the docs is hard
- Give a vision and communicate that lets others know where they are going
  - Roadmaps

**themaintainers.org**



- Info about how to get involved, setting expectations
- Notice if it is abandoned

Where do these moral obligations come from?

- This is different if you are paid vs unpaid
  - It isn't clear if the volunteer / nights & weekend maintainers have any
  - But they sign up for it
  - Example of npm stream, where a new maintainer added cryptomining. People blamed the original maintainer, "I don't know what to do"
  - Although "getting paid" is variable
    - Some people are just paid 10% by a company to just fix critical bugs, but others think they are being paid way more
  - There is an economic demand that generates these obligations that isn't going to pay maintainers of libraries that are relied upon.
    - Although kernel developers are paid a lot, some maintainers are
      - Although this is "contribution overflow" to own the roadmap (e.g. Docker)
        - This raises an obligation of the maintainers to act on behalf of their companies' interests
- There are people who sign up to be maintainers as aspirational for their careers
  - Mismatch between what people sign up for
  - Licenses say "As-Is" so why do people ask?
    - b/c it is built into the github interface?
- Difference at different scales.
  - A lot of OSS are small, but then they sometimes blow up
  - Buster Neece: most of his tweets are about the obligations he feels like others have taken over. Why do people think he has an obligation?
  - Is it ok to fork (hard fork) vs work to integrate upstream?
    - Causes more maintenance problems for other people, other devs who rely upstream, other users.
    - How do you know as a user which fork to use?
- This raises an obligation of the maintainers to act on behalf of their companies' interests
  - Assertion: If someone is paid to contribute on behalf of a company, it is no longer open source
    - Lots of pushback to this. Openly governed vs open source; free as in freedom vs beer.
    - E.g. Android AOSP vs Kubernetes,
    - Hard to think about maintenance / infrastructure under other political-economic systems. This only exists because we have a particular individual-corporate labor system. "The trains run on time"



- Software is a non-rivalrous good, can't fork a road.  
More incentive to maintain one built infrastructure than it is w/ digital.
- But the scarce resources are developer time, you could have this with a digital bridge design.
- Railroad tracks is where forks come from

Do you have obligations to people who don't know it is open source or contribute-able?

Actions: could create badges about how much commitment they sign up for

More granular positions: new triage role on github, day-to-day vs. long term teams

E.g. Binder has different teams where people can put themselves in more day-to-day vs long term roles

Tidelift is working on some things around this

What is the role for a maintainer emeritus

Although this makes us want more terms, because we compress this a lot

There is value in compressing all the work

Trademarking maintainer brands'

Scale is an issue, but also a lot of projects are relied upon but have few people. What is scale?

Debian MIA team -- checks in on maintainers

Topic Seed 2.0 : **What does it entail to be a maintainer?**

- Respond to Pull requests / etc.
- True as well in F/OSS as well as other Collaboratively driven Software Creation processes
- Clear/unclear - where do expectations derive from/ Ethos: no mess, go fix things / liabilities, being "on call" / no documentation - "read the code"
- reusability vs. usability
- Aspirational component / trade-off - is there informed consent?
- Buster Neece - Online Radio Software
- Visions seem to be important for the decision /
- Forking also has a cost in maintenance / end-users /
- "Liberation of innovation"
- Maintainers have an obligation in "mature" projects to formulate a Vision
- Compensation: does that change the obligation? / influence of upstream users
- Grantmakers view: funding mismatch - so dependencies need to be really clear / what is the public good in the maintainers time
- Economic demand creates the obligations, but never rewards the maintainer
- MPM
- "Contribution overflow"- Vision ("saying no is part of his job") - Governance question: dependencies
- "Spirit of the project gone" if closed company owns open source project- difference in maintenance



- Roads and bridges analogy: where does it hold up (on governance / orga / resources level) where doesn't it? (Software is a non-rivalrous good, can't fork a road. More incentive to maintain built infrastructure than it is w/ digital. )
- Issue count / react time- bad, bad metrics
- Maintainer emeritist:
- Stratified systems / hierarchy
- Trademark instead of License - different approach to IP
- Scale sets the difference (sometimes) / SQLite / Bootstrap / Debian MIA-Team

### **Unconference 3 Topic Seed: TBD**

(Mehdi Medjaoui (moderator), Don Goodman (moderator))

Text

ethics in FLOSS

<https://medium.com/doteveryone/https-medium-com-doteveryone-tech-workers-call-time-on-moving-fast-and-breaking-things-4bf62a90ef27>

<https://www.theiet.org/career/professional-registration/chartered-engineer/>

<https://www.tech-transformed.com/> - be a responsible innovator

<https://ethics.acm.org/>

Subsession 2:

#### **Do-ocracies as their discontents**

General premise in software (and especially OSS) is that you own it if you work on it

In some cases, software is not a do-ocracy. Like companies. But open governance vs open source.

There are benefits to do-ocracy in that you can go further with less discussion.

“Alone we go faster, together we go farther”

“Contribution overload” -- when companies contribute a lot to a project to set the vision.

Many companies are having issues around where their (OSS) projects are going.

Personal experiences with this:

- I am the sole maintainer working full-time, quit my job to do it. I'm spending the most time, so I should make all the decisions. But maybe I don't have all the info. Although lots of time, people just want things. But I'm actively trying to get people involved, but they don't show up.



- What if you make a mistake that makes it less popular? You can be a great programmer but a bad architect.
- It's hard to get people to care about anything other than code. Maybe I'm not the best at marketing or design or docs, but nobody was doing that.
- It is a very lonely task, like a lot of writing. This is also an issue with Awesome Lists.
  - Most projects are only maintained by one person.
  - Many projects are in different domains, but same stack and architecture. Want to put them into performance groups of the same size, age, stack, dependencies. Could you form a consortium to share community, techniques, issues?
    - Businesses do this, they set up realtors and furniture store owners
    - Mutual aid support groups, can share ideas, even maintenance burdens (e.g. taking time off)
    - ROpenSci does this for academic/scientific libraries
  - Also important to get out of your language / stack
    - Lots of language-independent issues, like triage, fundraising, newcomers
  - People often take over others projects
    - "It's not my baby" so I don't have as much emotional attachment to it, more willing to have others working on it
  - Writing is also like this
    - Writers have support groups
  - This is also way different in applications vs libraries
    - Applications get way less attention on GitHub
    - When your users are developers, this is way different
      - They know a lot more, but think they know more than they do
  - Developing software today requires far fewer people, but expectations are higher
    - More people work remote too
    - My job offered to pay 50% of my time to work on OSS, but it became more lonely when this happened, because nobody else was working on OSS. Hard to get people to give money. But with opencollective, gave a bunch of talks asking people/companies to donate.
      - More companies donate to open collective rather than individuals
      - Tidelift does this with their own salespeople, taking a cut. This maybe works better for projects that are small and don't need that much money. And how much do they decide to give each project?
      - Tidelift also plays a fiscal sponsor role, this costs money
      - Some people will work for less if they can do F/OSS (and live anywhere), others will not





- Although this track is the only one that has the luxury of career choice. Software is one of the only places where the maintainers condition is one of power, although this is less and less. OSS is undermining the power of SWEs in some sense.
- We're a mix of people who work at companies, universities, freelance
  - Also the model of governments funding projects as utilities
    - Pushback against this as “just a patch”, advocate instead for business-based monetizing solutions

Issues with problematic over-doers:

In companies, you can fire them. In OSS, it is harder.

Also a case where companies come in and overload a project's governance

Ownership:

Mindsets:

In a corporation, governance is owned by the one who own the project,

Babel- JavaScript -

Performance Groups

Support Groups

### **Software Daily Recap**

(Mehdi Medjaoui (moderator), Don Goodman (moderator))

Text

## **TUESDAY**

### **Unconference 4 Topic Seed: OSS as Critical Digital Infrastructure: Legal Technologies and Institutional Design (Panel)**

(Benedict Kingsbury, Thomas Streinz, Mehdi Medjaoui (moderator), Don Goodman (moderator))

Email: [guariniglobal@nyu.edu](mailto:guariniglobal@nyu.edu)

Twitter: [@guariniglobal](https://twitter.com/guariniglobal)

Website: <https://www.guariniglobal.org/critical-digital-infrastucture>



### Maintaining OSS:

Problem: Big companies are going to be using OSS more and more for their own uses. So much of business software sits on top of OSS that isn't well maintained

- pricing - we do want to incentives the companies using these tools to pay for them, because they're benefiting from them, and not contributing to their maintenance
- But the issue is if no one owns it, who takes the \$? Where does it go?

What is a commons that can lead to a support of tools (OSS), data?

How do we engage the people who have stakes in the OSS but who aren't the producers/developers

- Everyone should have some stake in how these things are built, but we don't have a way to do this. How do we even translate the tech stuff to non-tech folks who are affected by it, so they can have a say?

Example of how this can be problematic: MPM for Javascript - commercial service, VC backed entity

- you need a small script, can use someone else's. Import it, import all the things it depends on.
- left pad - left aligned numbers or something. Author got angry, pulled all his packages off the thing. It was very small. But many other things depended on it. So when this happened, lots of people couldn't update their software.

### Data and ownership, interoperability, transportability:

What tools do we have that may be able to help us to make data transportability transparent and democratic, and within the control of the people whom the data is about?

### APIs

- Is this a legal question? Not really, because the legal frameworks that apply, will depend on the context and the parties.
  - Example: If it's businesses working together, whatever contracts they draw up will govern this.
  - In this example, where businesses are collaborating, and there's data transfer, often there's strikingly little info on how the data can be used, and this might be that there are details that aren't public (problem for studying)
- Challenge for APIs, illustrated through an example of veterans trying to obtain benefits through US government structures:
  - veterans needing to share PII and PHI to get the healthcare they need - can't just do through the API, bc need authorization and access to be approved by each veteran



- Hoping that an API can be created to upload all data into system, using power of attorney - challenge is they're using legacy system on top of legacy system

### Legal Frameworks

- GDPR doesn't specify how data portability should be carried out (what's technically feasible is the standard, which is a moving target). More specificity in the law about how data portability should be handled
  - New CA law is copying GDPR in this aspect
- laws are not enough because the companies will make it difficult for you to access your data, because they don't actually want you doing it.
- Governments are constrained to geographical territories, insufficient for covering these issues
- Facebook and other tech companies using privacy law to stop researchers, etc from accessing even aggregated data.
- GDPR great, but seems to be trapped in this individualistic framework. Impact example - As an individual, I give permission to websites/apps to use my data, because I don't see what can be done with it, and I want access to the website. Would a more communally-oriented policy framework would be more impactful?
  - Is GDPR insufficient? outdated? Evolved from some more individual country's laws (Germany)
- India - new data protection policy, but also new way of regulating. Every citizen owns their own data. They want to prevent a situation where a marketplace is created for purchasing individuals' data, so they're thinking the government should be an intermediary

Part of the issue is that there's an asymmetric landscape. Tech companies have more power, and governments, international organizations follow their lead, and allow the companies to set the standards for how things are done. Anti-trust going to have to be part of the solution, but only part - in some instances, local governance processes have had success. Barcelona is an example.

What would we like to see?:

- Need alternative institutions that govern data sharing. Don't think our government is prepared for this at all, would need new
- Example: PHI - Obama admin put out guidelines for interoperability for health info. Traction just got started bc trump admin was like, this is IT, use this standard. Fire standard - everyone is starting to use it, happening. Totally baffling for even semi-technical people, not at all democratic, not cool. It is democratic in the sense that if you have that expertise, you can contribute to the protocol.
- Could we look at the history of the library system, and the Mark Standard, to see how we could solve the problem?:
  - Mark was decided in the late 60s. Evolved into different dialects, and not all are compatible. Purely syntactical systems fall into this trap (example: FITS), but those with semantic and pragmatic frames force choices at the beginning - pick your poison
- Is data a public good infrastructure?
  - Infrastructure is a body of relationships, data facilitates these relationships



- Payment service *something something* (*Couldn't type and listen*) in India
  - imposed neutrality governance. Users can share banking information
  - In this example, APIs are the pipes

## **Unconference 5 Topic Seed: Open Source as Liturgy (Presentation)**

(Henry Zhu, Mehdi Medjaoui (moderator), Don Goodman (moderator))

Email: [hi@henryzoo.com](mailto:hi@henryzoo.com), Twitter: [https://twitter.com/left\\_pad](https://twitter.com/left_pad), Website: <https://www.henryzoo.com>, happy to chat!

### **[Link to OSS as liturgy notes](#)**

*All participants have stated they are OK with this discussion having public notes.*

Liturgy: public worship rituals, meant to bring you together

We talk about belief/theory vs practice -- what practices are behind the beliefs.  
Also, what are OSS's rituals vs tech's rituals, are these different?

Religion has a sabbath day, should we have this in OSS? Or is the sabbath your OSS day from your work life? Do we need a sabbath from a sabbath?

In America: We have Thanksgiving, what is OSS Thanksgiving

And thanksgiving is followed by Black Friday, a very consumerist one

Birthdays -- should we celebrate project birthdays?

In OSS:

- Hacktoberfest
- Sprints
- Standup
- Getting your first PR merged -- like first communion / coming of age / PR mitzvah
- The pull request
- The release
- Onboarding / giving someone commit rights
  - Also handing off lead maintainer to the project
- Kicking people out / desysoping
- Forking a project (in the pre-GitHub sense)
  - Exit, Voice, and Loyalty
  - Reconciliation / the great merging
    - Node / IOJS



- Numpy
- Forking in the GitHub sense

[In ancient greece, liturgy was the public donation of goods by the wealthy.](#) So OSS is always this kind of work? Is OSS a vehicle for tech companies and/or employees to get rid of their guilt of privilege, as confession or a covering up? (FL)OSS as protest, corporate social responsibility, building alternative spaces.

There's the other side of this, people working in aspirational labor and possibly exploited

What if we think of more polytheistic or Eastern religious religious traditions? Individual vs community

A lot of OSS work is in India and perhaps other eastern cultures? is expressed through a community lens ironically through large megaCorps like Google, Amazon or Microsoft who form a large chunk of tech employment in the country whereas open source in the west has and is largely expressed through an individualistic lens, with certain assumptions

Vs.

How many people have the same ideas of OSS as a particular politics as you do?

Religious rituals can work to either smooth over those ideological differences or highlight them -- the canonical prayer recited by rote vs shaking hands during the peace

Rituals can help highlight invisible work (or obscure it!)

Cathedral vs bazaar -- there's a distinction in OSS that does have this religious metaphor

Ritual is also common outside of religion, it is any time there is a cycle

They are highly durable

Rituals maintain the software and the community

They do keep people some out, which is kind of their point

Rituals also are about keeping people out

Separation of religion into sacred vs profane, confession of sin

ritual of purity / openwashing

Celebrity maintainers as messiah / spiritual leaders, lots of charismatic authority. Any rituals to deal with this?

Apache Software Foundation -- it is a faceless entity, a tiny bit bureaucratic, but that is good



What new rituals do we need?

- Onboarding / mentoring / initiation -- not a lot of this in OSS
  - Although lots of mentees leave, this is frustrating
- Rituals of inclusion / karma
  - Writing code is commodified virtue, you see it in its results
  - There are lots of tasks where you often don't see anything back that benefits you
  - Common discourse in OSS of "scratching your own itch" which means you don't do things that don't benefit you as your own user (e.g. good design, docs)
  - "Being a maintainer is when you start caring about other people's problems, the project as a whole, or open source as a whole"
- Restraint / fasting (vs. feasting)
- Everyday personal rituals: making coffee, checking notifications
  - Shifts from coding to reviewing
- Guilt about the number of open issues

Contacts

- Henry Zhu - Topic creator/host! [@left\\_pad](#) - Podcast on [liturgy](#), [cities/liturgy](#).
- Varun Adibhatla - Protesting Smart City predatory tech with Open Source and data collaboratives @ [argolabs.org](#) - [Gnostic Internet](#)

### **Unconference 6 Topic Seed: Building a Bridge to a Legacy Application – How Hard Can that Be? (Presentation)**

(Scott Ford, Mehdi Medjaoui (moderator), Don Goodman (moderator))

Text

### **Unconference 7 Topic Seed: Designing APIs for Humans: Developer Experience (DX), User Experience (UX), and the Social Practices of Software Maintenance (Presentation)**

(Maria Vidart Delgado, Mehdi Medjaoui (moderator), Don Goodman (moderator))

Email: [maria@adhocteam.us](mailto:maria@adhocteam.us), Twitter: @mlvidart, Website: [www.marividart.com](http://www.marividart.com), GitHub: @mlvidart

Text

[Link to Developer Experience / Community notes](#)

#### **What is Developer Experience?**

- Something companies are starting to invest in
  - Helping make beautiful documentation
  - Building communities



- How do we know an API is good?
  - Community is a measure of success (engagement and feedback)

### **What is community for?**

- Some companies think community is a marketing thing, something they can point to
- The best communities are organic, people come together because they *care*
- Colloquially, a collection of people who feel a connection to one another

### **Is StackOverflow a community?**

- Maybe, for some people? Maybe traditional wisdom is "yes"?
- Definitely a tool for building community: companies sometimes spend time answering, building knowledge base?

### **When does community form?**

- Maybe when things are bad. When they need fixing, complaining.
- Or maybe when a need is being solved.

### **Experiences trying to manage communities**

- An experience managing API users in a Google Group
  - Questions of Policy or Technology come up
  - Lots of effort goes into distilling ambiguous upstream answers into clear downstream answers
- GitHub has good success with Campus Experts
  - Students are helped to go out and train, teach, talk
  - What makes students special?
- Auth0 has a developer program where they support training, teaching, talks

### **What are ways to monitor the health of communities?**

- How long it takes to get an answer to a question
- What material benefits do the participants receive?
  - Participations confers status in StackOverflow

### **Moderation and community trust**

- Codes of Conduct, Terms of Service, are the well-known tools for moderating
- Some communities have moderators who don't even answer questions, just moderate abuse
- With government, particularly tricky issues of trust

### **Is OSS a community?**

- It used to, perhaps, but it doesn't now. It's too diffuse and diverse.
- Is developer culture anti-social, against community building?

[Link to Burnout notes](#)



*All participants have agreed that notes can be taken.*

Lots of burnout as volunteers / activism. Lots of tech people do OSS or other things because they don't feel good about their jobs.

Feeling like you have to be 24/7, working weekends

If you're supporting people who are doing volunteer / nights and weekends, then you're doing that on nights and weekends -- who maintains the maintainers

Your goals mis/aligned w/ org

Often tech workers who volunteer for civic orgs, you may not get a lot back, treated as IT

Safety / toxicity

Not enough time

But this requires doing an inventory of everything and making hard decisions

Enjoyment / connection

Lots of companies like to say they value work/life balance, but really they don't. Management has to lead by example.

Schedule send so that you don't appear to be working long hours ("just timeshifting"), but this is hiding your bad work/life balance

In global teams and OSS, the sun never sets on the project. This can work well but also not so well.

Everyone has a different threshold, comfort level, life situation, desire for work/life balance

If we don't have social rewards for these things, it won't change

OSS as a "do-ology" leads to burnout, you sometimes have to be always on to lead

We celebrate the firefighters who pull heroic hours when things break (and not the people who ship clean code that doesn't break), this can be toxic.

What is the burnout-aware version of the blameless postmortem?

When someone does the 2am sacrifice, you tell them to take time off.

- There are often moments when heroic hours are needed. Dealing with sudden upstream / vendor changes that require major migration. How do you build a culture where, no matter what happens, you can't go over X hours, like pilots
  - Ex in aircraft crash on experimental plane -- it was the last test, everyone was tired. The thing that went wrong, everyone was trained to deal with. Need to have people who advocate for it when people say, "we can do just one more thing"





- It starts culturally so early, high school / undergrads pull all nighters and are proud of it. When students brag that they pulled all nighters, you shouldn't be proud of them. But this is celebrated in undergrad CS culture
  - Where does this come from?
    - Many cases where undergrads can't estimate how much time things take, they don't plan.
    - But is that a badge of courage or something you are ashamed of?
    - This is also huge in startup culture.
    - Also "unlimited vacation" is usually less vacation. Make not taking vacation a performance metric.
      - Also have to fit it in the structure / bus factor, can't have a team structure where people taking time off causes others major stress (or requires that you be called in when things burn down, because things will burn down)
      - Need a doc where people can say when they're off and what they might want to share, and others can leave you notes
    - Normalizing the out of office notice / away on slack
    - This is different in Europe, where August is off
  - People in second careers who come from bootcamps often take cultural norms from those from more traditional background. They are often placed on support roles for 1-2 years before getting on the engineering track. There was a hope they would bring healthier work/life balance norms and break down the upstairs/downstairs of engineering/support, but nope.
    - Lots of "voluntary" things like all night or weekend hackathons
- 
- Do companies pay the full cost of maintenance? Rarely
- Who gets promoted? / Career paths
  - Lots of programmers don't want to be managers
  - Peter principal: when you're competent, people keep you in your job, the failures go up. You also get people promoted to management b/c of good software skills
  - There is a recognition that software and manager is different tracks, often the way to get you into a higher salary band is to move to management
- Technical debt as burnout
  - Who gets to accumulate tech debt and who pays it down
- Slack
  - Being always on is the worst! Need to normalize the away. But you can't set out out of office.
  - Treat slack like email, but the whole point of slack is that it isn't
- Quitting stuff and saying no



- How to gracefully leave -- emotional things and practical things
  - Give the keys over, set an auto reply,
  - Build in that it is okay to abandon the project (esp in volunteer)
  - Death doula for volunteer orgs / farwell fund
  - Fixed term positions
- Making having children at events more normal

Stripe: used to celebrate shippers, now celebrate fixers

### **Unruly Bodies of Code in Time (Presentation)**

(Marisa Cohn, Mehdi Medjaoui (moderator), Don Goodman (moderator))

Text

### **Software Maintenance in Cyberspace Competition: The Geopolitical Functions of Slack (Presentation)**

(Bradley Fidler, Mehdi Medjaoui (moderator), Don Goodman (moderator))

Text

### **Unconference Recap**

(Mehdi Medjaoui (moderator), Don Goodman (moderator))

### **Software Track Reports Out**

#### **WEDNESDAY Report-Out Prompt**

Format of reports out:

5 min: reports/provocations

10 min: talk at tables

15 min: open discussion

**Don and Mehdi opening:** in contrast to other maintenance work, open source software maintenance work is done in public and related to prestige and reputation among professionals. This can result in burnout in a social media driven world. Often the work is carried out among the higher classes who have more free time. Thinking about the ethics of the maintenance of software: who uses your code? What obligations do maintainers have to the users of their software in light of complaints in light of software maintainers working on their free time? What can/should you do when code used maliciously? When do you retire a legacy system? How do you let go of it responsibly? Mehdi emphasized that it was a mix of academics and practitioners. As an unconference, folks picked their own topics, but we had technical topics, social human condition (such as burnout), and in the end they put proposals regarding regulation and entrepreneurship. Questions to provoke: what so much value is created but the maintainers don't see? What



does the fate of so many depend on so few? What is the practice, policy of care from culture of consumption to care--what does this look like?

**Open discussion:** The first table discussed that the conversation around software and maintenance is relatively advanced compared to other domains, and what lessons can be learned in light of that history as well as the social and financial capital built up. Also, in software, there are often higher paths for pay, which helps offset maintenance problems/burnout in other sectors, especially because there is a need for software maintainers that is valued and appreciated among organizational cultures. How do we maintain code in light of emphasis on creation rather than maintenance of the old? Thinking about *versions*, the issue of communication among developers/users is important, there might be rethinking of *numbering system*, to communicate to non-maintainers the level of work that goes into software maintenance. Someone else brought in how the software maintainers write the rules and also set boundaries for rewards, and how the “pain” is felt by those outside who use those systems. How do we think about criticality in terms of design and use? Another table spoke about the “governance regime” of norms of standards in software maintenance--this connects to open source hippocratic oath, but how can this be enforced? At least it’s a step forward. Also, self-policing among users and developers is its own kind of work. Kendra, from a lawyer perspective, spoke about “don’t be evil” clause, bringing in that licenses are very difficult to ensure compliance, and governance may be more relevant. Juliana brought in the idea of software maintenance as a labor of love among volunteers, and therefore likely to be underpaid or not paid, and what is the responsibility to keep it going or let die, especially when gets in the way of self-maintenance? What about maintenance work costs more than production of the new (comparing, say, the high cost of dry cleaning compared to buying new clothes)? Jess Farrell mentioned the challenge in some libraries, which are not even allowed to use open source software.

**Final thoughts:** Don mentioned the history of open source software, and think of themselves as apolitical, but people are seeing that these pieces fit together more realistically, in terms of licenses, to move into a new phase of collaborative software development and from an interdisciplinary perspective. Mehdi also mentioned the history of open source licenses. He also mentioned the sense of humor among open source software maintainers!