Andrew Harmel-Law 🏡 @al94781

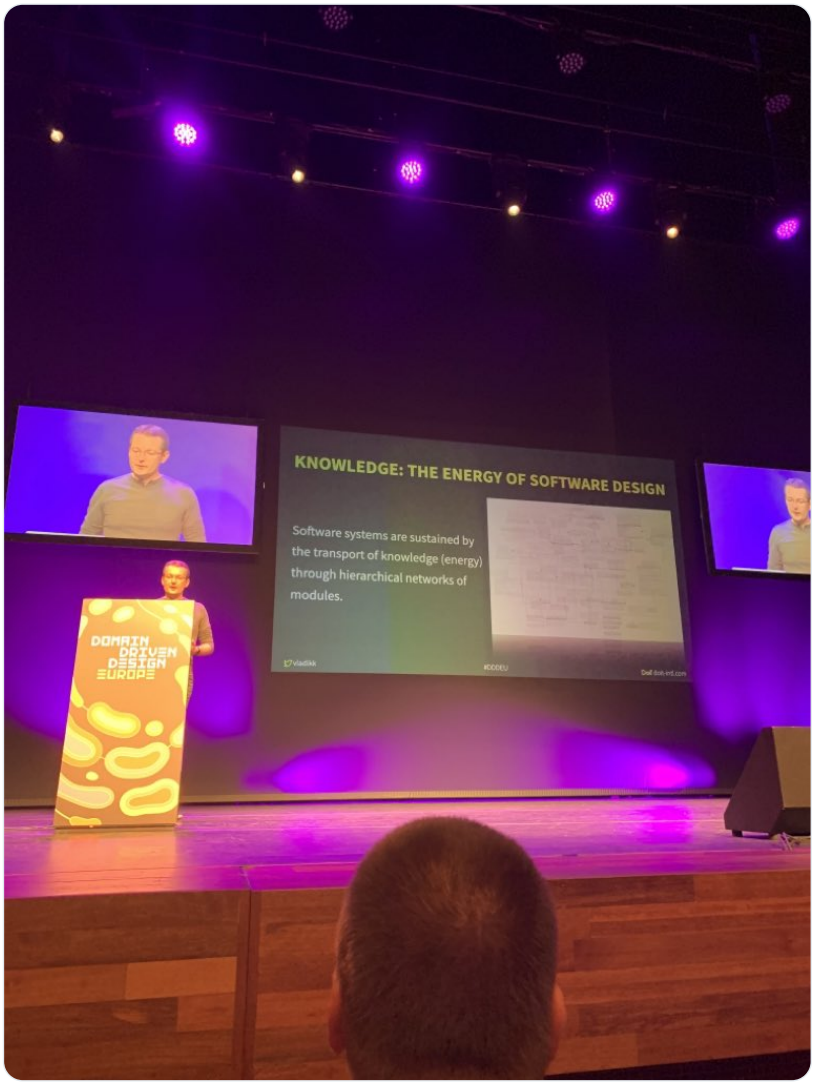
Aug 29 · 59 tweets · al94781/status/1564288746228416513

Thread🧵: Thoughts on the anthropology of software (power and freedom special edition.)

(Thanks to @dianamontalion as usual for inspiring it, and also this time making it coherent too. 🙏🙏🙏)

Software is a team sport, a collective endeavour, and it is increasingly accepted that human dynamics have as big an impact on software as anything else. (C.f. Conway, @TeamTopologies, @mtnygard's "HR do the first draft of your architecture" etc.)

Code is knowledge. Code is *literally* knowledge (and viewpoint) made manifest. (C.f. @vladikk's @ddd_eu 2022 keynote.)

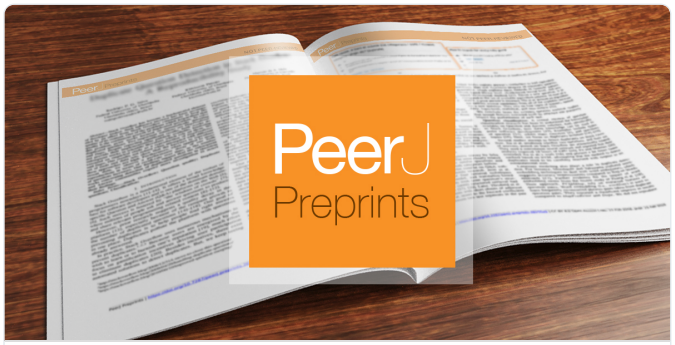


Not only that, code is history of knowledge. Code is a record of how a collection of people understood a problem over an extended period of time.

But code is also power, and the ability to write / approve / deploy code can be used for ill or for good, intentionally or unintentionally. (C.f. @ziobrando's "Dungeon Master" and The Phoenix Project's Brent).

In so doing, code *solidifies* these power structures; embedding that which is almost inevitably unequally distributed, either unintentionally or intentionally, but embedded all the same.

Don't believe me? Have a read of this (I'll wait for you...)



Gender differences and bias in open source: Pull request acceptance …
Biases against women in the workplace have been documented in a variety of studies. This paper presents the largest study to date on gender bias, where we compare acceptance rates of contributions fr…

https://peerj.com/preprints/1733/

Written code is therefore a geology (genealogy?) of these power structures right up to the present. Written code forms the landscape where records of our old ways of (dis)organising, and ideas that won, *continue* to shape everything because this code is where teams live *now*.

The form of the code is irrelevant. Even "no structure" code serves as both a record and a source of power. Hidden in the "Big Ball of Mud" paper (laputan.org/mud/) are the "experience", "visibility" and "complexity" forces which bring this about.

These sources of knowledge/power which produce the BBoM forces are *never* evenly distributed. When we look carefully at Big Balls of Mud, we can *always* see this manifest.

Let's take this a step further. Have a read of Jo Freeman's famous paper "The Tyranny of Structurelessness" (https://www.jofreeman.com/joreen/tyranny.htm)

When Jo talks about "elites" (frequently "friendship groups") which function as "networks of communication" she could be talking about groups collaborating to deliver code.

She makes it abundantly clear that power structures are ALWAYS present. They might not be formal, but they restrict options, and they restrict who can exercise them.

So let's be clear, "BBoM code" isn't always written by accident - it's not always due to of lack of skill or bad luck.

Perhaps instead the authors were forced into it by the structures of power around them.

Or perhaps they smudged it all up intentionally, to ossify an imbalance of power, and therefore knowledge, in their favour.

We all experience this “code-as-knowledge-and-power” force every day. We feel our freedom to act being restricted or permitted. (Note: this last point is key, what is restrictive for one might well be enabling for another.)

Can we dissect what is at play here? Is it possible to consider what is happening with “code-as-knowledge-and-power” and work with it intentionally, instead of suffering/benefitting disproportionately at the hands of it?

@davidgraeber and @davidwengrow can help here. In their book “The Dawn of Everything” (

The Dawn of Everything: A New History of Humanity
Read 1,492 reviews from the world’s largest community for readers. A dramatically new understanding of human history, challenging our most fundamental assu…
https://www.goodreads.com/book/show/56269264-the-dawn-of-everything

) they postulate three prerequisites for power, any power, to become established:

1. Control of violence
2. Control of information
3. Individual charisma

At first glance these seem worlds away from software, but if we imagine for a second that to write code is the ultimate freedom (because code is, after all, pure knowledge and power) what might we see?

A small digression is in order…

Frequently when embarking individually on code-work (before we humans combine and try to work together) our code-freedom is absolute. Working 100% on our own means can make all the decisions. Anything is possible. (It might not be advised, but it is possible.)

Moving to working as a group inevitably curtails these freedoms; typically increasingly so as any form of surrounding organisation becomes more established.

These organisational means of curtailing freedoms - either formal or informal, but always necessary for collaborative work - can be co-created by all, or envisioned by a blessed one/few.

The adoption of these means by the group can be trustingly accepted, or arbitrarily/forcibly imposed.

And the means-adoption may or may not include the mechanism for the group to update the means themselves.

Each of these aspects is an exercise of a form of power. In Wengrow and Graeber's terminology, the power here is the "control of violence" and it comes from those who specify the means and mode of organising and deciding.

If an individual does not follow this mode they will be formally or informally punished by the group (punishment here can go right up to dismissal).

In software, this exercising of "controlled violence" - defining and shaping ways of organising and deciding - directly impacts the code: because the team's structures and ways of working translates directly *into* code.

This is not to say that this "organising" is de-facto bad. As I already said, to co-operate as groups then there is always a need for some kind of control. It is the definition and exercising of this control where issues can arise.

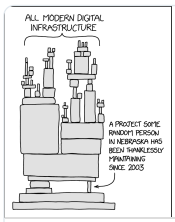
Let's pull out again. What we just saw was one form, but Wengrow and Graeber identified three. What about the other two?

Well, working with code itself conveys power, a power that remains (largely) in the hands of individuals irrespective of their control of the means of organising. This power is Wengrow& Graebers "control of information" & it either amplifies, or acts contrary to, the other powers

To control information, according to Graeber and Wengrow, it needs to be unequally distributed. Now we know code is knowledge, but knowledge (and code) need not be self-evident and accessible to the many.

Code's ability to misdirect and mislead is legendary.

(How often have we seen code written by a long-lost colleague which is still exercising power, and shaping activity, long after the author is departed?



Dependency

https://xkcd.com/2347/

)

This ability of code to mislead manifests at *all* levels.

It can be an attribute with an opaque or inconsistent name.

It can be a function which doesn't do what it says it will, or does more than it ought.

It can be a module which does too many things, or half does-many things without competent doing anything.

Or it can be an architecture which screams "JavaEE" and nothing about what it actually does.

These powers are *so* available to us as developers that we exercise them accidentally _all the time_. Even this conveys power. Power to prevent understanding. And power to prevent change (where power to prevent something getting to PROD is the greatest of these).

Think: If you are one of the few (the only one?) who understands and can manipulate a piece of code, you are depended upon, right? And this is _irrespective_ of where you sit in any organisational structure (almost inevitably a hierarchy).

Given these facts, why don't we see information-power imbalances wielded _everywhere_?

(It cannot simply be because so many of us are committed to the group and it's collective goals.)

This consideration brings us to the third Wengrow / Graeber prerequisite of (unequally distributed) power (structures): Charisma.

Let's remind ourselves quickly of the three prerequisites for power imbalances to become established:

1. Control of violence
2. Control of information
3. Individual charisma

I failed to mention earlier one key additional fact (ok, I did it on purpose).

Graeber and Wengrow posit that for power imbalances to begin to stick, any *two* of these three need to be in place. That's where, for us, charisma comes bursting in.

Charisma can manifest as organisational position and status (more on which in a few tweets) but it can also, if strong enough, undermine positional power too.

Why? Because charisma-power manifests in the code. We've all worked with (against? despite?) code we're scared to change because we don't understand it.

But who here has worked with code we're scared to change _because of who wrote it_.

All code which we can't (or won't) change has power.

But code with "charisma-power" combined with one of the two other powers, is well on its way to becoming immovable, at least in the absence of incredible force majeure.

Let's take one last step back. How do outside (societal/systemic) power structures play into this?

We already know forms of organisation mirror societal "norms". This privileges some people over others. Why would it not manifest directly (consciously or unconsciously) in code?

whispers: it totally does. And not just when we call things "master"/"slave" and "blacklist"/"whitelist", "girlfriend-/mother-test". Code is knowledge, remember? And knowledge encodes bias after bias.

We've got enough in mind now to try and conclude.

What does all this mean for organisations which want power and autonomy / self-management held more strongly by the collective than a sub-group or individual?

It means we need to be aware of these three forces, and how they are establishing and intersecting, *all the time*.

To do this we need to enlist the help of *everyone*. That means listening to everyone, and valuing/balancing what they say.

And my "everyone" here is broad. I mean those with hyper-detailed, hyper local viewpoints, as well as broad, longer-term, systemic perspectives.

All the players in our #sociotechinical ecosystem. (Paging @trondhjort 📣)

Why "everyone"?

Because if you're not being consciously and terribly inclusive, you are letting your biases seep into everything again via (in)conscious exclusion.

It also means we need to foster awareness of group(s), formal and informal, and their *systematic* relationships with both the code and each other.

We humans and our code are intimately coupled, so signals from *both* are incredibly, and equally important.

What might those signals be? Well, Wengrow and Graeber say, they are emitted when people report any of their three "freedoms" being curtailed or exercised *unequally*:

1. "Freedom to move" - to change / add / remove boundaries; to head off somewhere independently and set up something new.

2. "Freedom to disobey orders" - to not follow a general consensus, or the overall design, or principles of the overall architecture; to code for oneself in a way which is contrary to the general expectations of the whole (n.b. This is NOT permission to be a jerk.)

3. “Freedom to reorganise social relations” - Sociotechnical. Architecture. Is. A. Thing. (And Conway, etc etc.) It is how we interrelate. It is the roles we play. It is what we expect of others and what they expect of us. This freedom is the freedom to craft these relationships

This freedom is the freedom to craft these relationships *for ourselves*.

(Let me repeat once more for the record, freedom-curtailing of these sorts are *essential* for any form of collaboration. You need to enquire into the specific circumstances.)

If we watch out for, and value and inquire into, any signals of these essential freedoms being curtailed *at the expense of the overall, collective organisational goal* then we begin to inoculate ourselves against creeping power imbalances.

And that is a good thing, for all of us.

And for all our code.

• • •