

# HR Has Concerns About My Git Strategy

A journal note on why politics apps, repo cleanup, deploys, and self-respect should not be treated as one task.

2026.04.06 17:21 DENNIS HEDEGREEN JOURNAL OPEN-NOTE

[HTTPS://HEDEGREENRESEARCH.COM/ARTICLES/HR-HAS-CONCERNS-ABOUT-MY-GIT-STRATEGY/](https://hedegreenresearch.com/articles/hr-has-concerns-about-my-git-strategy/)

---

HR has reviewed the events surrounding the politics apps, the mirror repos, and the late-night belief that one more push would probably fix it.

HR has concerns.

Since HR in this case also means Hedegreen Research, this should be treated as both a personnel matter and a systems note.

The review finds that one employee attempted to hold the following in his head at the same time:

- Norway as the next internal politics adapter
- a cleaner multi-country engine
- broken Denmark and Sweden public apps
- GitHub repos that were not set up cleanly enough
- Streamlit deploys
- public doors on the site
- the emotional desire to end the day with visible proof that the day had counted

Human Resources would like to note that this is too many layers for one nervous system.

The employee would like to note that each individual layer was, in isolation, completely defensible.

That is exactly how these things become stupid.

No single task looked absurd.

The Norway work made sense.

The engine refactor made sense.

Fixing the public apps made sense.

Cleaning up GitHub made sense.

Making the mirrors honest made sense.

Deploying cleaner public doors made sense.

What did not make sense was trying to treat all of them as one continuous act of seriousness.

That is the detail HR keeps circling in red.

Pressure in software rarely arrives dressed as danger.

The room is quiet.

The machine accepts input.

The terminal stays available.

Git keeps taking commands.

The browser still has a refresh button.

Nothing in the environment tells you that your judgment has started to downgrade itself in the background.

So you improvise a moral story instead.

You tell yourself you are being committed.

You tell yourself you are closing loops.

You tell yourself that visible persistence is a virtue.

Sometimes it is.

Sometimes it is just escalation with a nicer biography.

HR believes the sentence "I am still making progress" should be treated with suspicion once a person has started confusing:

- architecture with rollout
- motion with resolution
- available tools with permission
- one more push with actual calibration

There is a specifically modern dignity trap in this.

When the work is physical, exhaustion becomes harder to romanticize.

The body gets obvious.

The material pushes back.

Something in the room starts saying no.

Computer work is more polite than that.

It lets you keep getting worse in private.

That is one reason political software is such a bad substance for this kind of day.

It comes preloaded with importance.

Everything around it sounds consequential:

- election data
- public method
- deployment
- source of truth
- live apps
- public trust

All of that is real.

But if you are not careful, the seriousness of the subject starts covering for the sloppiness of the state you are in.

That was part of yesterday's problem.

Not that the politics apps were fake.

Not that the architecture was fake.

Not that the repairs were unnecessary.

The problem was that I started acting as if every unresolved thing had become one unified emergency.

It had not.

Some things needed a fix.

Some things needed sequencing.

Some things needed documentation.

Some things needed a new repo.

Some things needed a new URL.

Some things needed sleep.

These are not the same category of work, no matter how emotionally efficient it feels to call all of them momentum.

HR has additional concerns about the phrase "while I am here I might as well..."

The review finds that this phrase recently expanded to include:

- while I am here I might as well fix the rerun bug
- while I am here I might as well clean the repo structure
- while I am here I might as well create the right branches
- while I am here I might as well update GitHub
- while I am here I might as well deploy new Streamlit apps
- while I am here I might as well update the public doors

This is not workflow.

This is how a person turns one solved problem into six active fronts and then calls the resulting stress focus because the font in the terminal still looks calm.

The humiliating part is that the systems note was true before the joke was.

The apps really did need calmer sequencing.

The mirror model really did need to be written down.

The public doors really did need to be updated only after live verification.

That is what makes this kind of overload dangerous.

It is not nonsense.

It is too much truth in one container.

Now that the apps are stable again, the repos are cleaner, the Streamlit doors are updated, and the public links point where they should point, the whole situation suddenly looks almost reasonable.

That is a relief.

It is also a warning.

The bad day tried very hard to present itself as a test of persistence.

In retrospect it was more like a test of whether I could still distinguish between:

- a real blocker
- a messy state
- a missing document
- a deploy step
- and my own need for the day to end with a certificate of usefulness

Today the answer is better than it was yesterday.

That is not because I became wise overnight.

It is because the systems are quiet enough again that I can hear the distinctions.

HR therefore recommends the following:

- the politics apps should be allowed to live for a bit
- the mirrors should be treated as mirrors
- the engine should remain the source of truth
- Norway should wait for the next calm pass instead of being dragged in by momentum
- the employee should stop trying to convert strain into virtue

Management accepts these recommendations.

Fortunately management is also me.

That arrangement has weaknesses.

But today, for once, it has produced a useful memo.

---

RELATION MEMORY

MATURES FROM [Don't Push When You Push Too Hard](#) · Turns the general stop-rule into a concrete source-control and recovery problem.

SYSTEM REQUIREMENT [The Tool Learned to Say Not Yet](#) · Git strategy is part of the same operating rule: the system should stop bad actions before they become costly.

NEARBY [Publishing Became a Command](#) · Repeatable publishing needs version discipline underneath it.

