

Don't Push When You Push Too Hard

Software work makes false urgency feel like responsibility. Sometimes the right technical decision is to stop before force turns into worse judgment.

2026.04.06 00:29 DENNIS HEDEGREEN OPEN OPEN-NOTE

[HTTPS://HEDEGREENRESEARCH.COM/ARTICLES/DONT-PUSH-WHEN-YOU-PUSH-TOO-HARD/](https://hedegreenresearch.com/articles/dont-push-when-you-push-too-hard/)

I sat at the computer all day trying to make a political app work.

I also wanted to get better at GitHub.

Better at pushing properly.

Better at closing the loop instead of leaving work half-real.

The worst part is that the direction was not even wrong.

The Norwegian app made sense.

The shared motor made sense.

The multi-country direction made sense.

I was still working on the Norwegian election-data side internally.

The motor probably should work across all three.

But seeing the right architecture is not the same as having earned the rollout.

I was trying to update Denmark and Sweden into a multi-country frame before the underlying state was calm enough to deserve it.

That was the real mistake.

Not wrong direction.

Wrong speed.

The public surface was already telling the truth in its own way:
not functioning reliably right now.

And internally there were already real movements.

A fix here.

A committed adapter change there.

A Norwegian branch that still made architectural sense.

That is what makes this kind of day dangerous.

Not total chaos.

Just enough partial correctness to keep you pressing.

Computer work is especially dangerous here because the room stays calm.

Nothing catches fire.

No machine visibly bends.

No material gets used up in front of you.

You can keep trying for hours and call it discipline because the damage is not theatrical.

But the damage is still real.

Your reading gets worse.

Your pattern recognition gets worse.

Your sense of proportion gets worse.

You stop distinguishing between:

- hard problem
- bad state
- bad implementation
- bad judgment

That is the threshold that matters.

Not the moment where you feel bad.

The moment where you become less trustworthy to your own work.

There is a strange moral confusion in software around this.

Shipping matters, so stopping starts to look indulgent.

A repo can always take another commit, so another commit starts to look useful.

Push is a valid technical action, so pushing harder starts to feel like the right human action too.

That is false.

The world survives a day without your app.

That sentence sounds soft.

It is not.

It is a correction against false urgency.

Most of the time, what is actually threatened is not the world.

Not the users.

Not history.

What is threatened is your desire to finish the thing today.

Those are not the same.

And once they get confused, force starts to masquerade as seriousness.

You say you are being committed.

You may just be grinding your own judgment down past the point where it can still help.

Do not push when you push too hard.

Do not send the commit just because you have been sitting there for ten hours and need the day to mean something.

Do not keep forcing the system just because you are emotionally overinvested in ending the day with visible proof.

Sometimes the correct push is tomorrow.

Sometimes the most technical thing you can do is admit that the controller is no longer calibrated.

This is not anti-work.

It is not anti-persistence.

Hard things do require long effort.

But long effort only helps while you can still tell what kind of problem you are in.

Once that goes, the quality of the work changes.

Not because the app became impossible.

Because you stopped being the right instrument for it that day.

That is harder to admit in software than it should be.

I wish I were the kind of person who could stop.

The machine still accepts input.

Git still accepts pushes.

The screen still looks available.

So you mistake availability for permission.

But not every open terminal is asking for more force.

Some are asking you to come back tomorrow less stupid.

— Dennis Hedegreen, still checking

RELATION MEMORY

RETURNS TO [Publishing Became a Command](#) · Command-speed creates the need for a stop rule when pressure starts to damage judgement.

NEARBY [A Tool Can Be Right and Wrong at the Same Time](#) · Both articles separate a useful tool from the false confidence that can surround it.

NEARBY [HR Has Concerns About My Git Strategy](#) · The practical version of the same pressure boundary appears in Git and release discipline.

