

This is a continuation of the [Four Factors of an Effective Software Development Pipeline](#), where we will focus on factor two: **Skip The Gatekeeper**. Check out [Factor 1: Separate Build, Provision & Deploy](#) if you missed it.

## Factor #2: Skip the Gatekeeper

*In [Norse mythology](#), **Heimdallr** is a [god](#) who possesses the resounding horn [Gjallarhorn](#), owns the golden-maned horse [Gulltoppr](#), has gold teeth, and is the son of [Nine Mothers](#). Heimdallr is attested as possessing foreknowledge, keen eyesight and hearing, and keeps watch for the onset of [Ragnarök](#) while drinking fine [mead](#) in his dwelling [Himinbjörg](#), located where the burning rainbow bridge [Bifröst](#) meets heaven.*

source: <https://en.wikipedia.org/wiki/Heimdallr>

And yet still people deny that life imitates art? How many teams still have “release engineers” or “system administrators” whose job is essentially to be a modern day Heimdallr? They are there to click the button to production and otherwise watch out for Ragnarök before, during, and after.

Skipping the gatekeeper is the evolution of moving beyond this practice, and it begins with trust.

### Trust

I frequently talk about culture and organizational development on LinkedIn and here on this website, and I'll try not to fall into that diverging topic too much, but we must first be willing to admit that trust is an essential component of any successful software development team. If you cannot trust *anyone* on your team to deploy to production, I believe you have a fundamental cultural issue; or perhaps worse, a lack of skills. Either (or both) should be mitigated/resolved as your #1 priority if you are in that situation.

With trust, you can safely assume *and expect* that anyone on your team can deploy to any environment whenever necessary. The sheer gain in flexibility and efficiency for your team with this alone will pay for itself in no time.

## Accountability

[Self preservation](#) is, according to Wikipedia, universal among all living things. This includes software engineers!

When a production deployment is handled by someone else, and the immediate fire(s) are also being put out by someone other than the engineer(s) who wrote the code, there is a separation of accountability that is inherent in the organizational structure that process enables.

If instead the engineer(s) are deploying their own code, you will see an irreversible and immediate improvement in the quality of the code being deployed because the same engineer(s) will do what is in their best interest because they are immediately accountable for their own work.

On a more positive note, this also enables and encourage more pride in the work being completed which is far more of a positive outcome than the negative reinforcement of making your team responsible for any failings in their work. The successes are also felt and they're far more immediate and encouraging.

## Speed & Efficiency

One of the most obvious results of removing your teams Heimdall is an immediate improvement in the speed and efficiency of deploying code. Business leaders win because they see completed work sooner. Customers win because their requests are fulfilled faster and more precisely. Engineers win because they are enabled to complete their work more rapidly and efficiently, giving them far more [purpose and play which leads to a higher ToMo](#); and the gains go on and on!

## Wrapping Up

When you have trust and accountability on your teams, an immediate improvement in speed and efficiency amongst your engineers is an inevitability that your entire organization will benefit from!

Recapping

Previously we discussed the first factor, which is to [Separate Build, Provision & Deploy](#).

Next: Pipeline Automation Should be Free!

The next factor is to [Pipeline Automation Should be Free!](#)

Share this:

- [LinkedIn](#)
- [Twitter](#)
- [Email](#)
- [Print](#)