Going from quarterly releases to weekly in a small startup company

Daniel Paulus, VP of Engineering @ Checkly

Gartner

COOL VENDOR 2023

At Checkly, we are a DORA Elite Performer

This **talk** is mostly **about** my **past experiences**

Seeing the **immense value of synthetic monitoring** as a driver for DevOps maturity was part of the reason I joined Checkly back in 2021

At Checkly I am fortunate to have a great team and **we operate at elite level** when it comes to DORA metrics

Still, **I use the DORA metrics as my compass** when making engineering decisions, to make sure we stay sharp

Before we start

As a technical leader I think the **DORA metrics** are very relevant as a tool to guide my actions.

I decided to take my team from manual, error prone, monthly to quarterly releases **to stable weekly deploys in weeks** rather than months.

Let me show you how I did it and **increased dev velocity by 70%** along the way in a ~40 people engineering team startup.

DORA metrics - Recap

Lead Time for Changes - The amount of time it takes for code to get into production

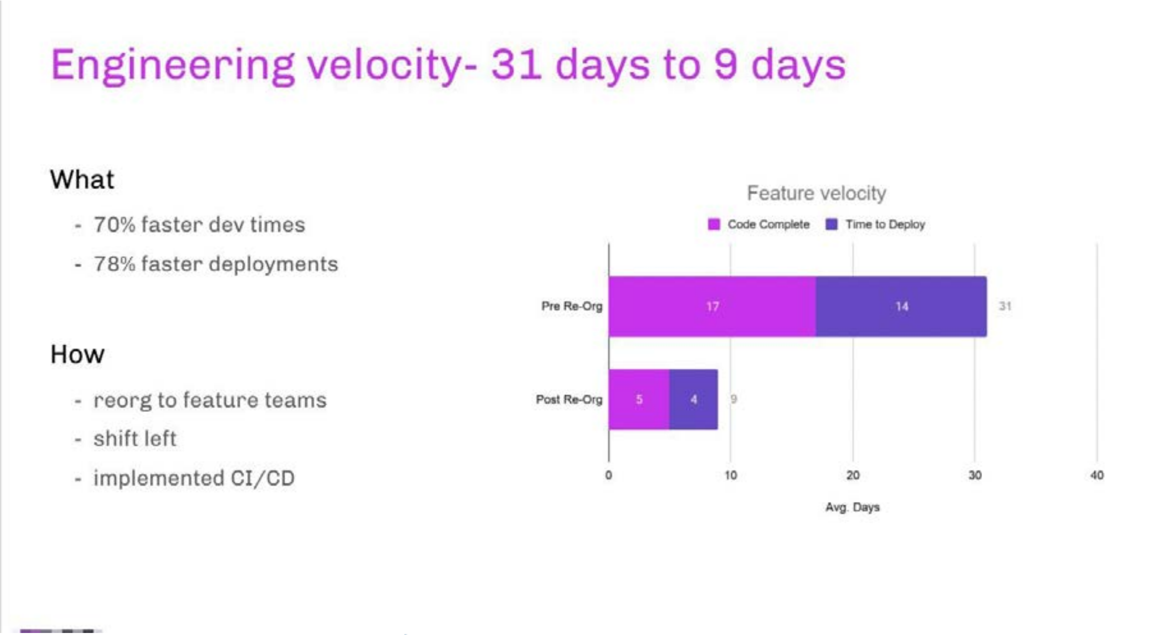
Deployment Frequency - How often an organization successfully releases to production

Time to Restore Service - How long it takes an organization to recover from a failure in production

Change Failure Rate - The percentage of deployments causing a failure in production

(https://cloud.google.com/blog/products/devops-sre/using-the-four-keys-to-measure-your-devops-performance)

How do I know? We had our program manager measure it!



Engineering velocity- 31 days to 9 days

What

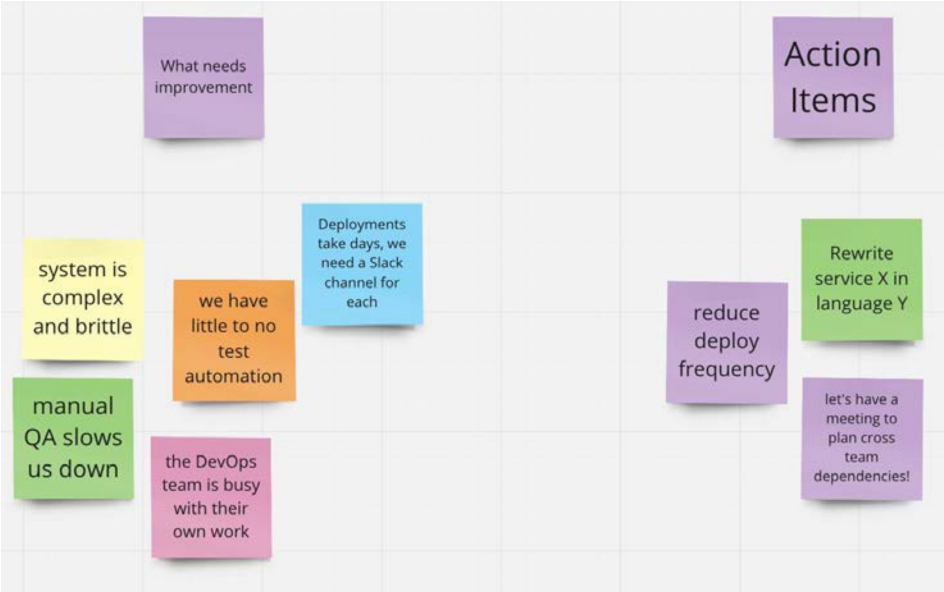
- 70% faster dev times
- 78% faster deployments

How

- reorg to feature teams
- shift left
- implemented CI/CD

Feature velocity

Have you ever been in one of these retros?



What needs improvement

system is complex and brittle

manual QA slows us down

we have little to no test automation

the DevOps team is busy with their own work

Deployments take days, we need a Slack channel for each

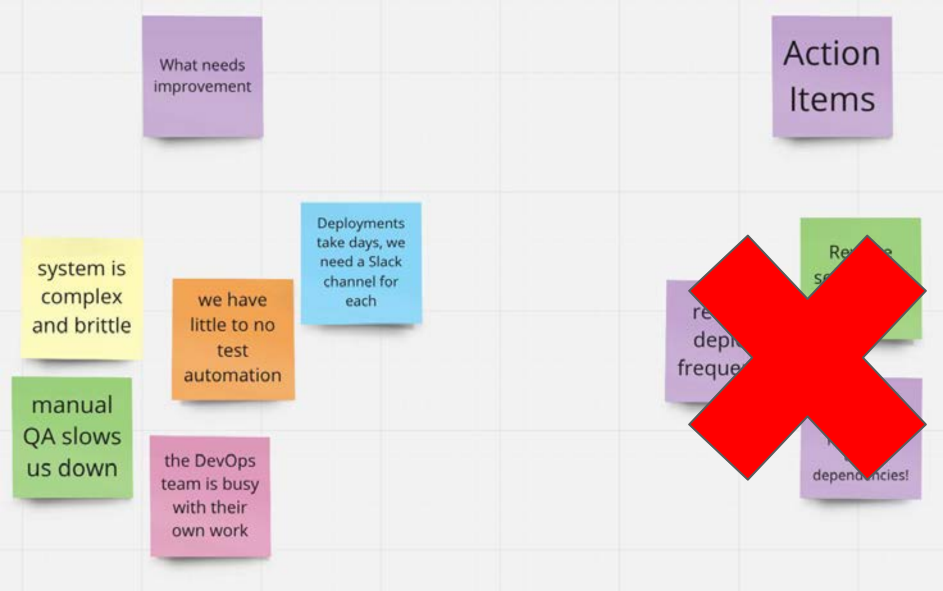
Action Items

reduce deploy frequency

Rewrite service X in language Y

let's have a meeting to plan cross team dependencies!

Have you ever been in one of these retros?



What needs improvement

Action Items

system is complex and brittle

we have little to no test automation

Deployments take days, we need a Slack channel for each

manual QA slows us down

the DevOps team is busy with their own work

re depl freque

Re e s

dependencies!

Don’t despair, use DevOps to guide your actions

There a couple of actions you can take, that will improve things.

Of course, this will be a tough journey and many people will have to be convinced.

A few words on 🧑🏿‍🤝‍🧑🏼Culture👭🧑🏾‍🤝‍🧑🏾 and Change Management

Enter every discussion with **patience, listen to concerns, take them seriously**, and explain the **purpose** as often as people need to hear it.

Some might leave, but you can eventually win over (most) people’s hearts

“Do we really have to deploy all services every week? Why?”

“Deploying often does not make sense, you only deploy when it is done”

“I feel we don’t need DevOps engineers anymore”

Difficulty: Easy to Hard 😆

Select a difficulty level:

Concrete Steps To Take - Tech (easy)

Convince developers to build ci pipelines on Github Actions or similar.

Automation is key for cycle time.

Simplify, **Simplify**, Simplify and.. Simplify!

Carelessly introduced **Complexity is the root of all evil**.

Reduce branches, delete unused code, remove unused dependencies

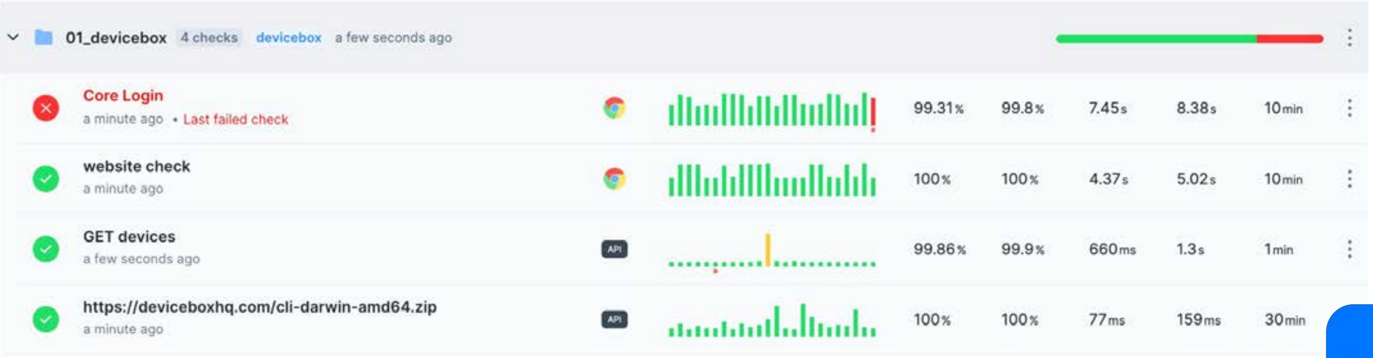
Concrete Steps To Take - Tech (easy)

Use **synthetic monitoring**! It's perfect for such a scenario.

Use existing or new e2e testing a **strong signal** if things work - in days.

Focus: get back to boosting DORA!

😭 We broke the login 😭



01_devicebox 4 checks devicebox a few seconds ago

✗	Core Login a minute ago • Last failed check		99.31%	99.8%	7.45s	8.38s	10min
✓	website check a minute ago		100%	100%	4.37s	5.02s	10min
✓	GET devices a few seconds ago	API	99.86%	99.9%	660ms	1.3s	1min
✓	https://deviceboxhq.com/cli-darwin-amd64.zip a minute ago	API	100%	100%	77ms	159ms	30min

Let's roll back 💪

Concrete Steps To Take - Tech (medium)

Automate also deployment pipelines

Introduce **zero downtime deployable** services

2x Simplify, **Simplify**, Simplify and.. Simplify!

Carelessly introduced **Complexity is the root of all evil**.

remove redundant dependencies, update versions, delete unused configuration and feature flags....

Concrete Steps To Take - Tech (hard)

3x Simplify, **Simplify**, Simplify and.. Simplify!

Carelessly introduced **Complexity is the root of all evil**.

Merge codebases, remove services, remove “DevOps” overengineering, simplify code

Concrete Steps To Take - People (hard)

Share past experiences if you have them from when you have seen this work

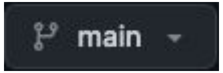
Build confidence through renowned material like the Google DevOps blog

Constantly explain, show and **lead by example**

Concrete Steps To Take - Process (hard)

Move to **trunk based development**

Have **PRs** and **Code Review**

Merge to  main and **keep it stable** at all times

Rollback fast if needed, fix bugs offline

Concrete Steps To Take - Process (hard)

Leverage your changes and move to **weekly** deploys

It hurts to do in the beginning, so you need to **do it more often**!

Difficulty: Emotional Damage

(Thanks https://www.youtube.com/@StevenHe for some really funny memes and videos)

Concrete Steps To Take - Organizational Change

Cross functional teams & You build it you run it!

No “DevOps” teams

Test Automation and exploratory testing!

No manual QA as a release gate

Thanks for listening!

Ping me on LinkedIn or the Checkly Slack Community if you have questions about DevOps or Synthetic Monitoring
Bye Bye 👋

Daniel Paulus, VP of Engineering @ Checkly

Gartner
COOL
VENDOR
2023