

# Data-driven software engineering at Fortnox



**01**

**Henrik Edegård**

**Development Manager**

**Developer Experience**

# 01

**Fun Fact:**

```
#: Update sales_table set  
firstname='henrik',  
lastname='edegard',  
street='myroad';
```

**#: Affected rows: 1 500 000**

**#: ... Panic...**

**02**

**630 000**

**02**

**Developer 1:**

**- Im tired of  
'npm update'**

**02**

# **Developer 2:**

- I know a  
tool...**

**02**

**Developer 1 :**

**- Wow, this  
solved *my*  
problem**

**02**

# **Developer 1 to Manager 1**

- Check this out,  
we all need  
this**



**02**

# **Manager 1 to “Man-in-charge”**

**- We should all  
do this**

**02**

**“Man-in-charge”  
to rest of  
Managers:**

**- Please do this**

**02**

**“Man-in-charge”  
to rest of  
Managers:**

**- Please do this**

**02**

# **Random Developer:**

- I get swamped  
with PR:s**

# 02

## Random Developer 2:

- **Hey, I made this script for chrome to hide your Renovate PR:s**

# 03

## So....?

Did anyone ask for data or questioned why

A developer initiated projects often misses looking at data or defining metrics for success.

# 03

# Data

Starting digging into data.

What goal should we have with a project like this?

- Reduced number of CVE:s in deps?
- More time spent on business logic?

Lets do a educated guess

# 03

## The data...

50 teams

500 developers

500 Repositories



# 03

## The data...

Renovate Pull Requests in 2 years

25 000 opened

21 000 merged

4 000 Declined

Avg time to merge: 12 days (at 8h per day)

# 03

## The data...

With 21 000 merged

at ~3 approves per PR

Equals to 63 000 Approves

at ~10 minutes of work per approve

TADA: 630 000 minutes of developer time.

(10 500h / 1 312 working days/ 5,6 years)

# 03

## The data...

5.6 years of work over 24 months

Are 2.8 full time employees

And what was the outcome?

# 03

## The data...

Probably more up to date dependencies

More secure?

Probably, but did you measure it?

# 03

## The data...

Did we get any better?

- How can you decide if you did not have any metrics?

Tweaks that can be made

- group patch/minor to one PR
- reduce when to run (once a week)

# 03

## Not the data...

CI success merge automatically

- Sometimes policies can say that this is not possible

CI success but our services failed anyway

- You need to invest in better testing

# 03

## Not the data...

This case is probably both a success and a failure

“The fastest way to get behind is by doing nothing when the world is rapidly moving forward”

So...at least do something. And if you set your metrics in the beginning, you can probably make say if it is a success or failure

# END

## My take away...

Before it spirals out of control...

Discuss pain points with your engineers on a regular basis

Demand a few key metrics for any project

- Both success and guard metrics

Democratize data, it is not there for supervising, but for improving. And make sure everyone knows this