

**What if we could build our software like...  
...we build our cars?**

(by using robots to automate all the things)

# We build our cars using robots cause they...

ensure high quality & are less expensive & free highly qualified worker.

**...but we still "build" our software by hand / using manual labor:**

what	% of our time
deployment	18
fix broken deployments	12
run (integration) tests	33
talk to the users to explain why something is broken	2
coding new features - create value	35 :-)

# WHAT MAKES THE ROBOT IN SOFTWARE PRODUCTION?

**CICD:** Continuous Integration + Continuous Deployment = Continuous Delivery

- Continuous Integration: build, integrate and test the software automatically
- Continuous Deployment: ship the software and their dependencies (e.g. infrastructure) automatically
- Continuous Delivery: continuous flow of new value with high quality to the customer

And as we are not perfect:

**MONITORING:** extract, collect and observe metrics - about everything

# benefits...

*We will ship faster, more reliable, with higher quality and at less costs. And you can trust our software products like our customers trust our cars.*

## ...for the DevOps teams

Like in vehicle production the use of robots has a direct impact on the (line) worker:

- the human employees could do other creative tasks
  - implement new features instead of doing robo-tasks
  - saving money by improving processes
  - think about / plan new features
- ...and at the same time employees will be more happy
  - no boring tasks & no avoidable mistakes (...uhh, blame culture...)
  - may increase motivation and reduce fluctuation

# HOW MUCH WILL IT COST?

- there will be an initial peak
  - developers might need to learn a few new tricks
    - most of us are highly motivated cause it will free us from rob0-task
  - we need to implement a few basic things
    - bit of infrastructure
    - an automation pipeline that models our production process
- ... so we think four man weeks over the next three months and a dedicated CICD server will do.
- in the long term we save as soon as we freed Devs from running tests or deployments manually - every seconds counts!

# implementation road map

We will not be able to stop the bleeding right now but

- we will define, monitor and report metrics to show the speedup and improvements we achieved
- we can setup the basic infrastructure pretty fast as we use standard tools
- we know the tasks that causing the most errors or take much time: automate first to save time and money now.
- align new development to new CI/CD strategy, e.g. lets start to implement new integration tests so we can run them automatically: do it once - do it right.
- we will select a few Devs to do deep dives on the required topics so they can guide the initial implementation and spread the knowledge by doing
- instrument any new feature with metric exporters so we can use them now to do manual checks and later in our automated telemetry setup

**LETS AUTOMATE ALL THE THINGS!**

**LETS AUTOMATE ALL THE THINGS!**

**LETS AUTOMATE ALL THE THINGS!**

**LETS AUTOMATE ALL THE THINGS!**

**LETS AUTOMATE ALL THE THINGS!**

**LETS AUTOMATE ALL THE THINGS!**