Continuous Delivery: Myths and Realities

Mario Fernandez

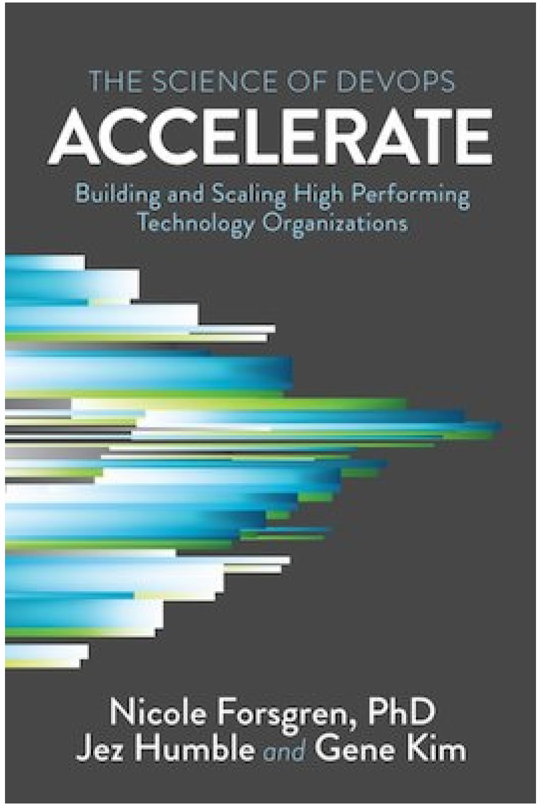
Lead Developer

ThoughtWorks

What is Continuous Delivery?

> Continuous Delivery is the ability to get changes of all types—including new features, configuration changes, bug fixes and experiments—into production, or into the hands of users, safely and quickly in a sustainable way.

Why Continuous Delivery?



thoughtworks.com/radar/techniques/fourkeymetrics

How to implement it?

5 Principles

Build quality in

Build quality in

Work in small batches

Build quality in

Work in small batches

Automation

Build quality in

Work in small batches

Automation

Continuous improvement

Build quality in

Work in small batches

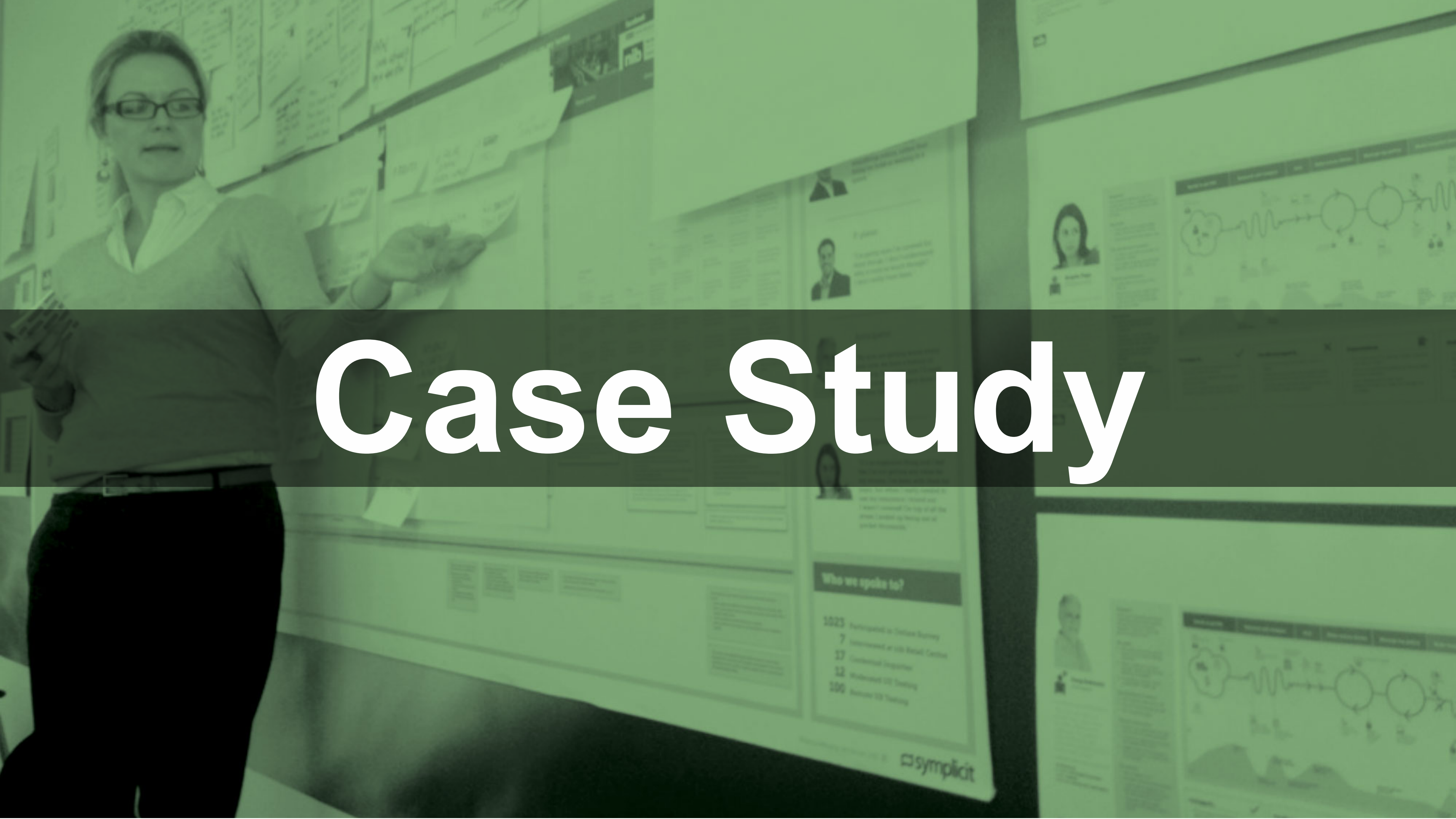
Automation

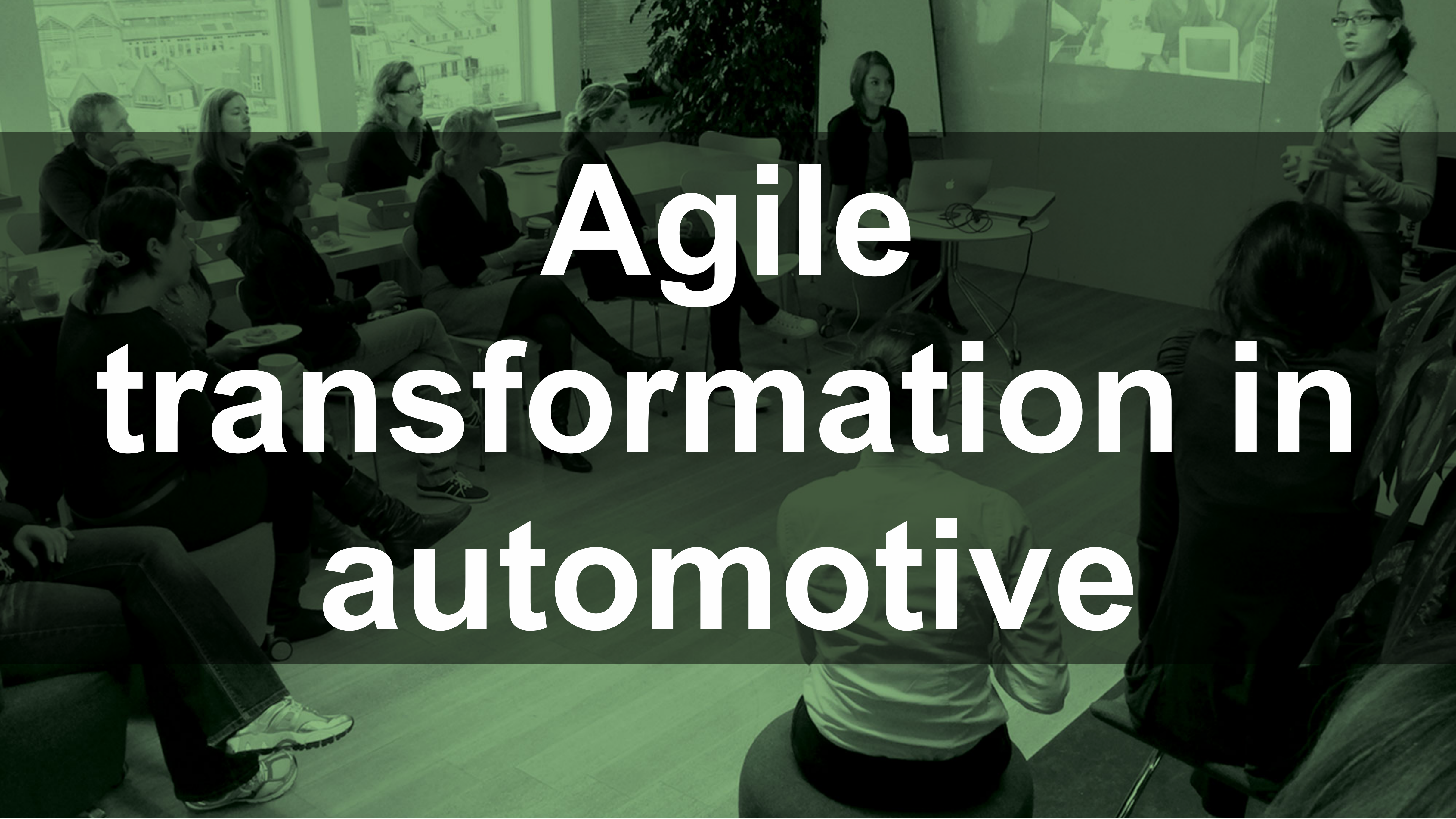
Continuous improvement

Shared responsibility

Let's try to get practical

Case Study

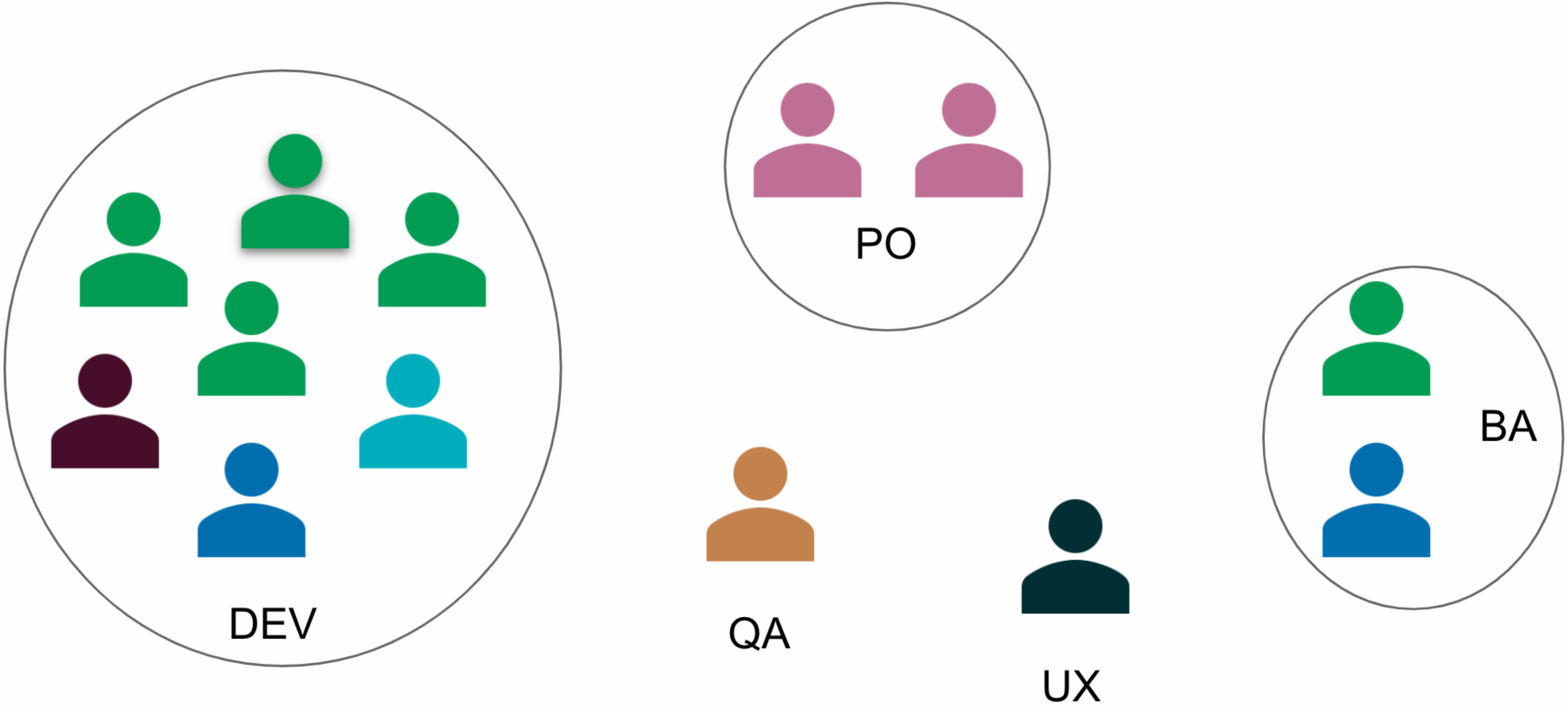




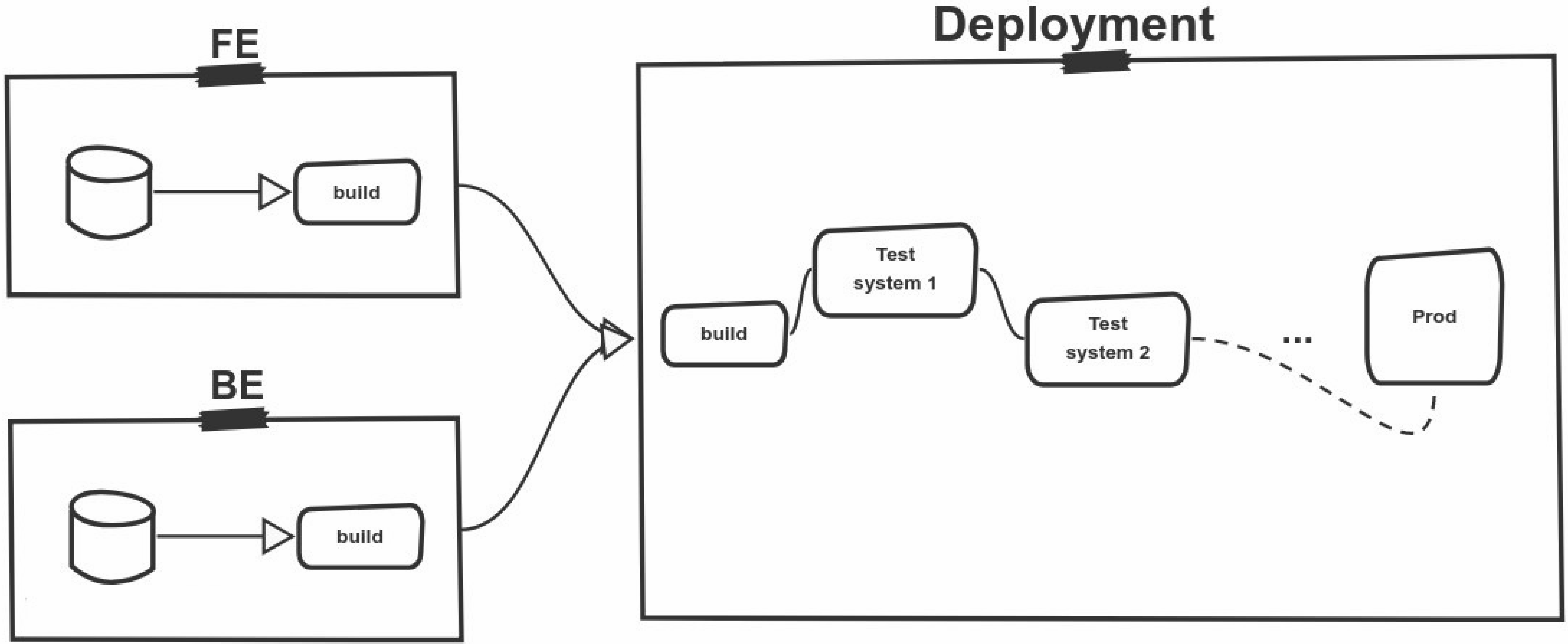
Agile transformation in automotive

Starting point

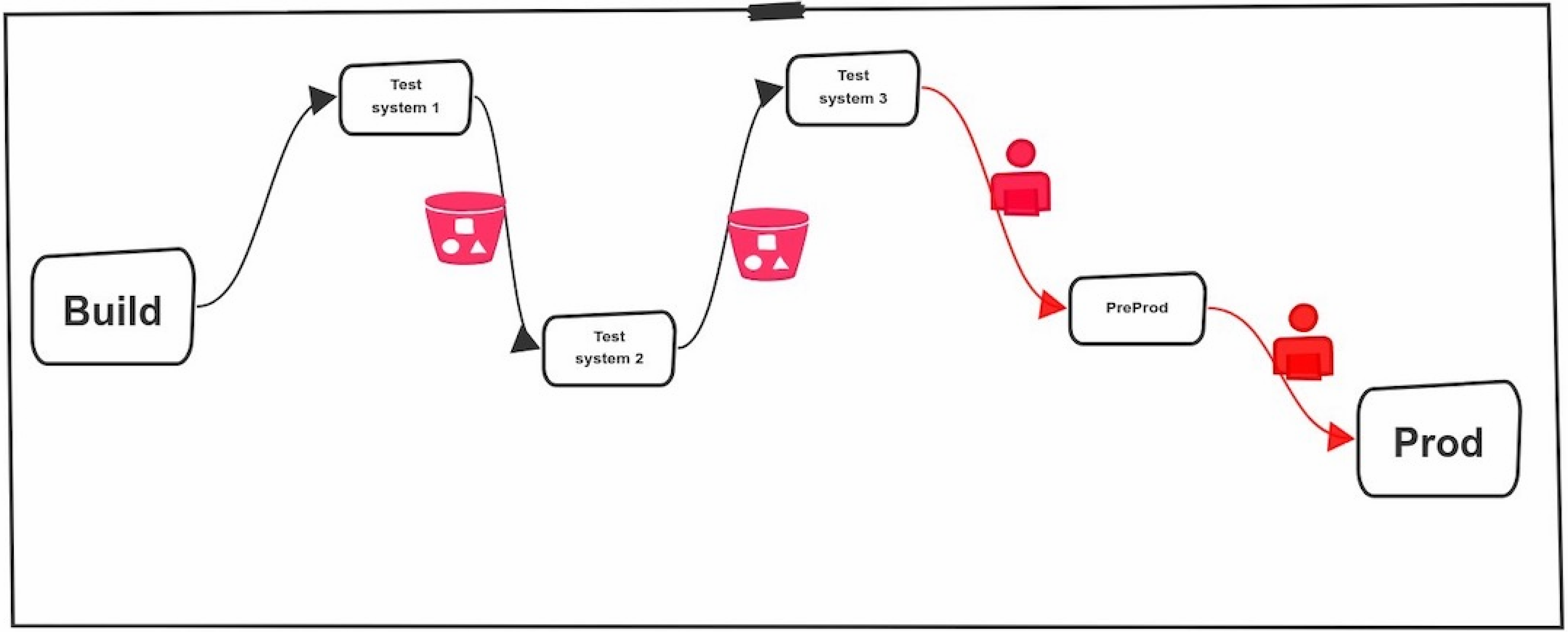
March, 2018


PO
DEV
QA
UX
BA

Multiple microservices with frontend and backend


FE
build
BE
build
Deployment
build
Test system 1
Test system 2
...
Prod

Deployment



Build

Test system 1

Test system 2

Test system 3

PreProd

Prod

THIS IS FINE.

Deployments every 2-4 weeks

Deployments every 2-4 weeks

Manual regression testing before each deployment

Deployments every 2-4 weeks

Manual regression testing before each deployment

Many open bugs

Deployments every 2-4 weeks

Manual regression testing before each deployment

Many open bugs

Unpredictable cadence

Deployments every 2-4 weeks

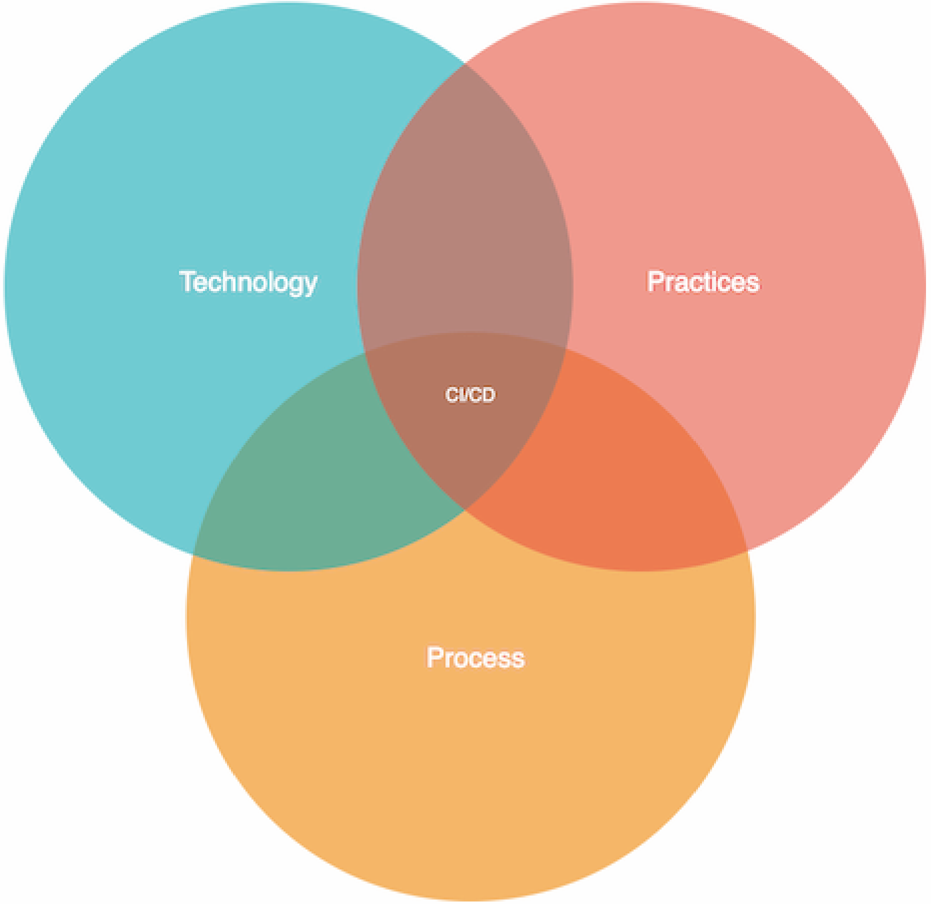
Manual regression testing before each deployment

Many open bugs

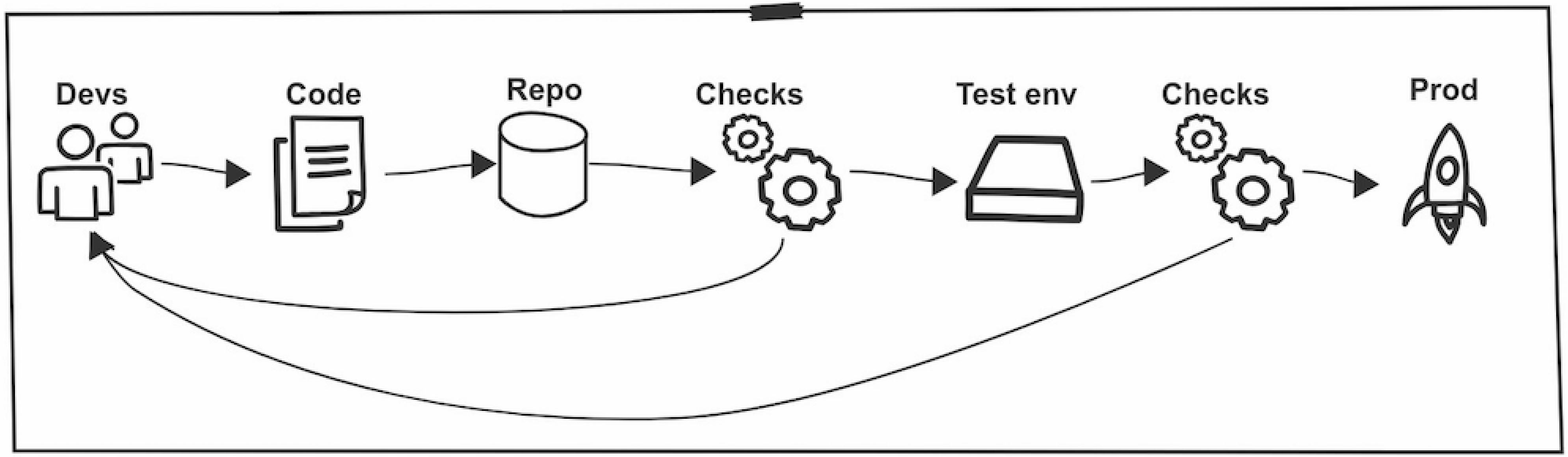
Unpredictable cadence

Regular delays

Something had to change


Technology
Practices
CI/CD
Process

Path to production



Devs

Code

Repo

Checks

Test env

Checks

Prod

Delivery pipeline

1

The code for the path to production is as important as the regular code

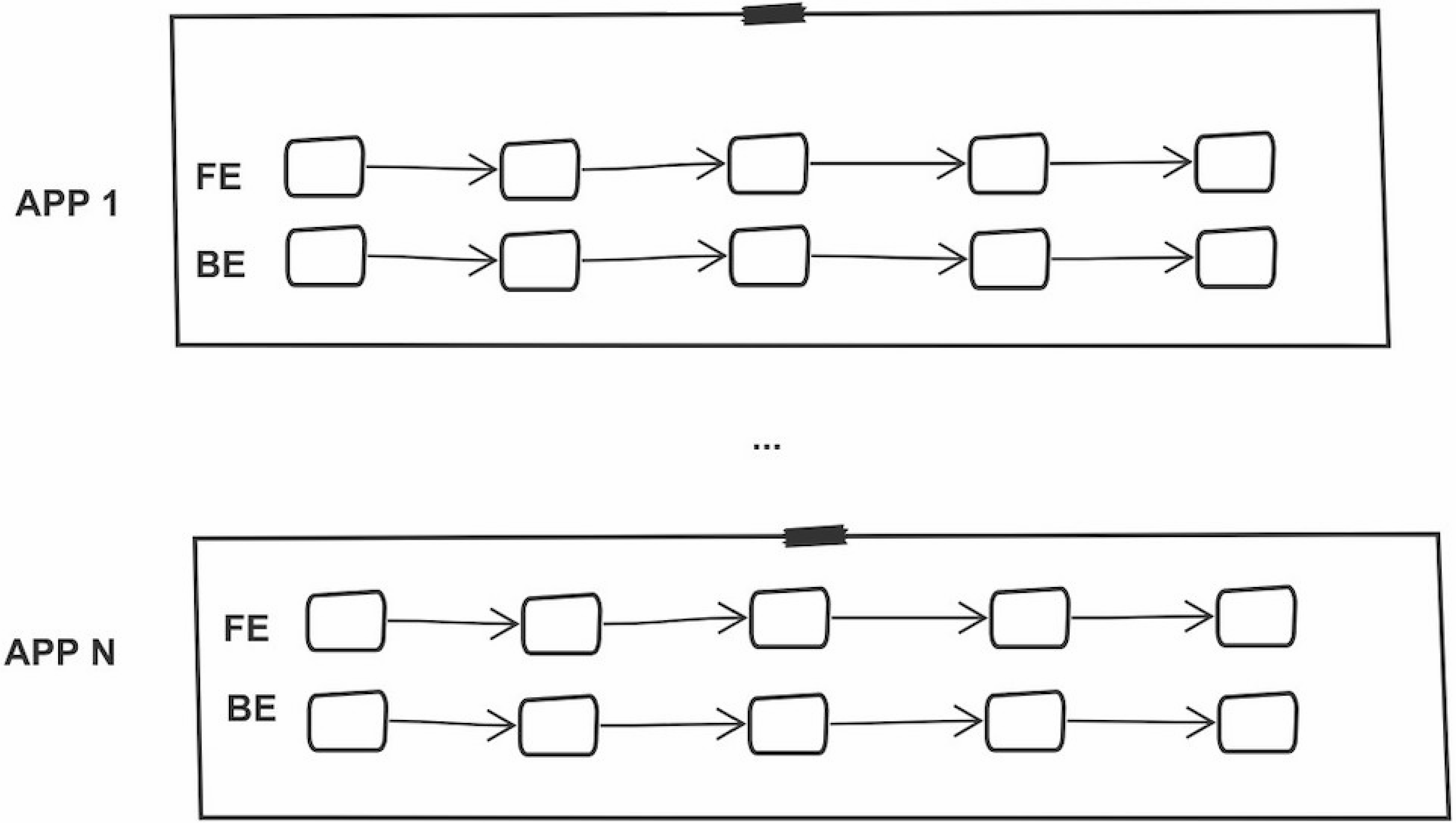
A good pipeline is code

www.gocd.org/2017/05/02/what-does-pipelines-as-code-really-mean/

```yaml
- name: test
  serial: true
  plan:
  - aggregate:
    - get: git
      passed: [prepare]
      trigger: true
    - get: dev-container
      passed: [prepare]
  - task: test-js
    image: dev-container
    params:
      <<: *common-params
      TARGET: js
    file: git/pipeline/tasks/tests/task.yml
```

```
platform: linux
inputs:
  - name: git
caches:
  - path: git/node_modules
params:
  CI: true
  NPM_TOKEN:
  TARGET:
run:
  path: sh
  dir: git
  args:
  - -ec
  - |
    ../shared-tasks/scripts/install-yarn-packages.sh
    ./go test-${TARGET}
```

Small, independent pipelines

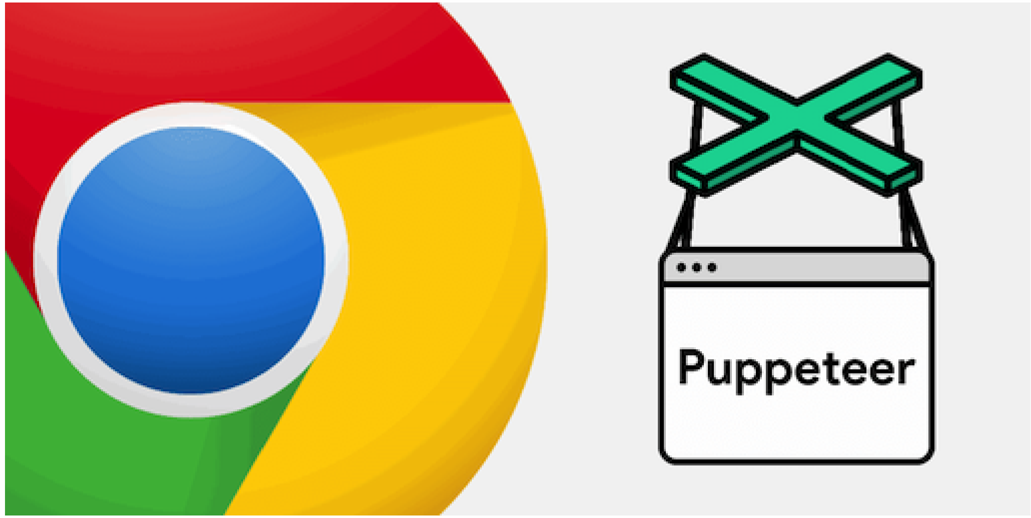

APP 1
FE
BE
...
APP N
FE
BE

Let the tools help you

ESLint

Prettier

stylelint


Puppeteer

thoughtworks.com/insights/blog/modernizing-your-build-pipelines

Infrastructure

2

How do you become faster by *adding* responsibilities to the team?

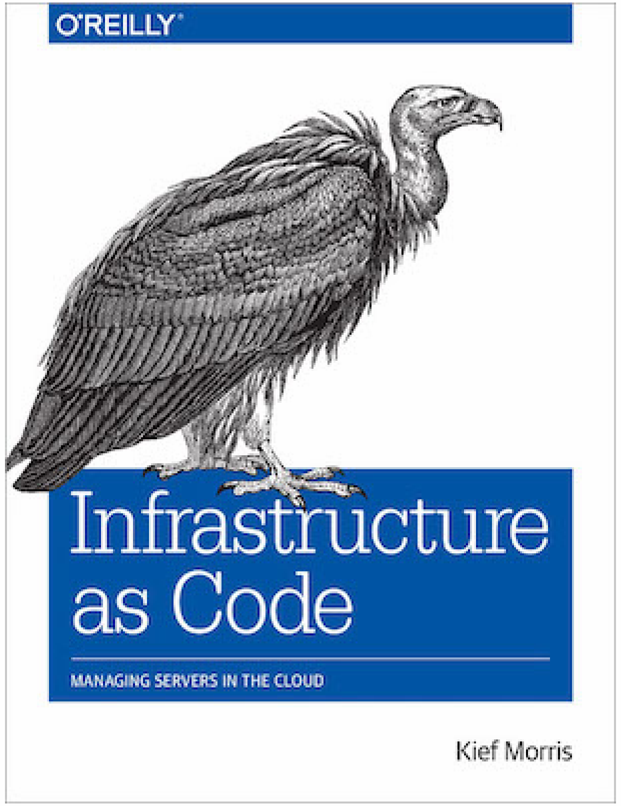
DevOps mindset

Autonomy

Agility

Leverage a larger community

Yeah, but how?



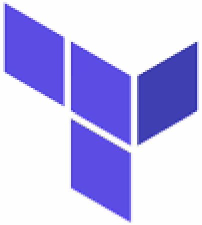
O'REILLY

Infrastructure as Code

MANAGING SERVERS IN THE CLOUD

Kief Morris

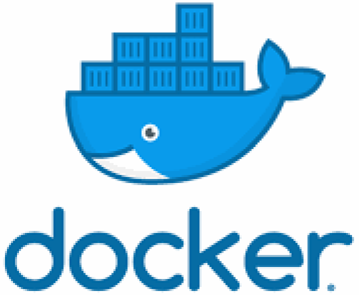
infrastructure-as-code.com

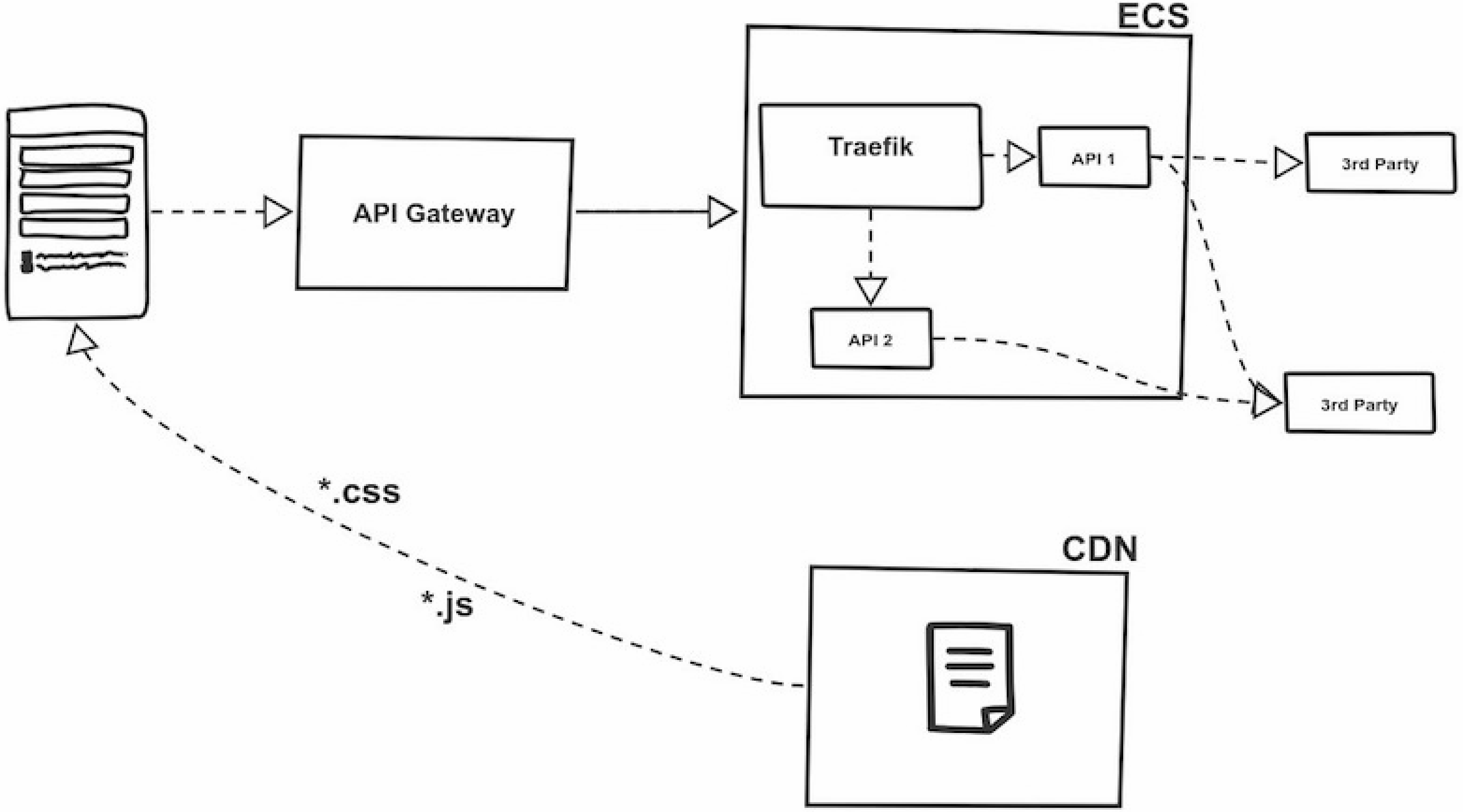


HashiCorp

Terraform

```
resource "aws_ecs_cluster" "ecs-cluster" {
  name = "${var.ecs-cluster-name}"
}

resource "aws_autoscaling_group" "ecs-autoscaling-group" {
  name                      = "ecs-asg"
  launch_configuration      = "${aws_launch_configuration.config.name}"
  max_size                  = "${var.max-instance-size}"
  min_size                  = "${var.min-instance-size}"
}

resource "aws_launch_configuration" "config" {
  name_prefix          = "ecs-launch-configuration-"
  image_id             = "${data.aws_ami.latest_ecs_ami.id}"
}
```


docker


API Gateway
ECS
Traefik
API 1
API 2
3rd Party
3rd Party
*.css
*.js
CDN

Support hero

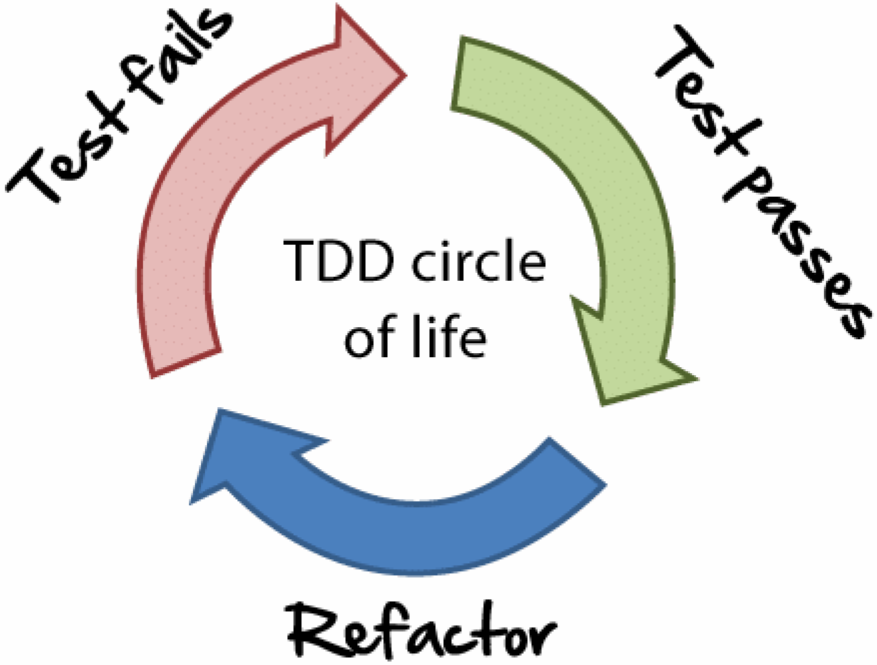
Owning your infrastructure makes it exponentially easier to deliver changes to your application

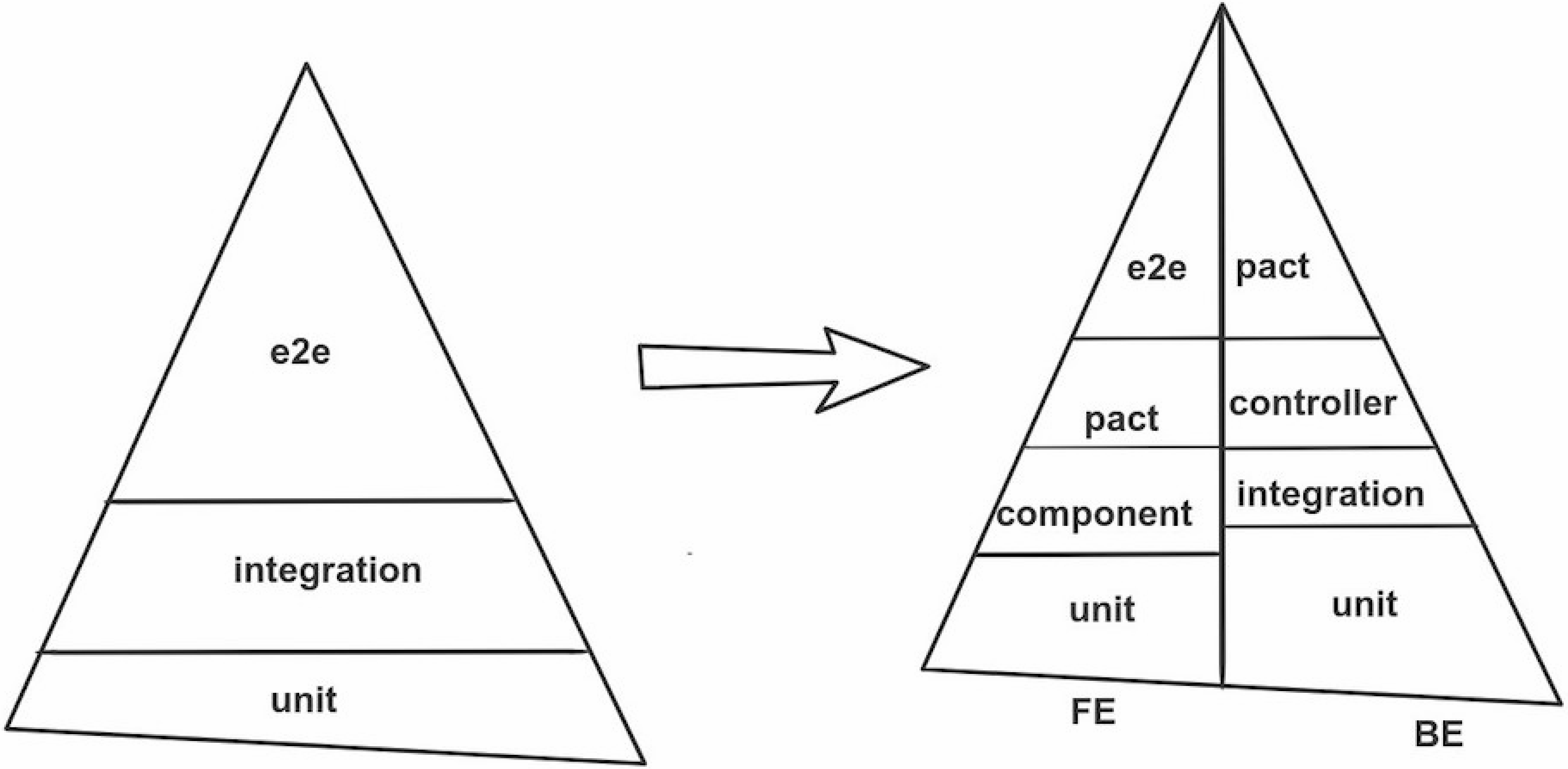
The Code

3

Design for Testability

TDD


Test fails
Test passes
TDD circle of life
Refactor



martinfowler.com/articles/practical-test-pyramid.html

TBD

A deployment is *not* a release

Feature Toggles

martinfowler.com/articles/feature-toggles.html

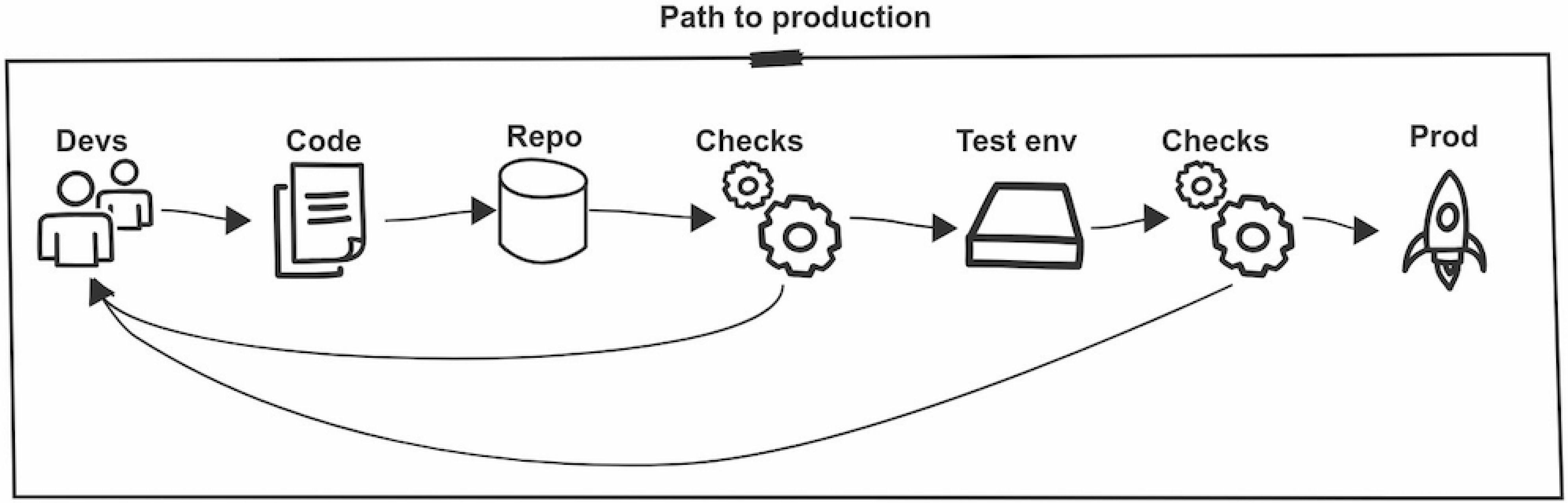
```
<section class="container">
  {{ service.label }}

  <app-feature-toggle-component featureToggleName="priceTag">
    <offer-price
      class="checkbox__label checkbox__label--inverted"
      [offerPrice]="service.price"
      [offerCurrency]="service.currency"
    ></offer-price>

  </app-feature-toggle-component>
</section>
```

```
@RestController
@RequestMapping(PATH, consumes = [MediaType.APPLICATION_JSON_VALUE])
@ConditionalOnExpression("\${pact.enabled:true}")
class PactController(val repository: Repository)
```

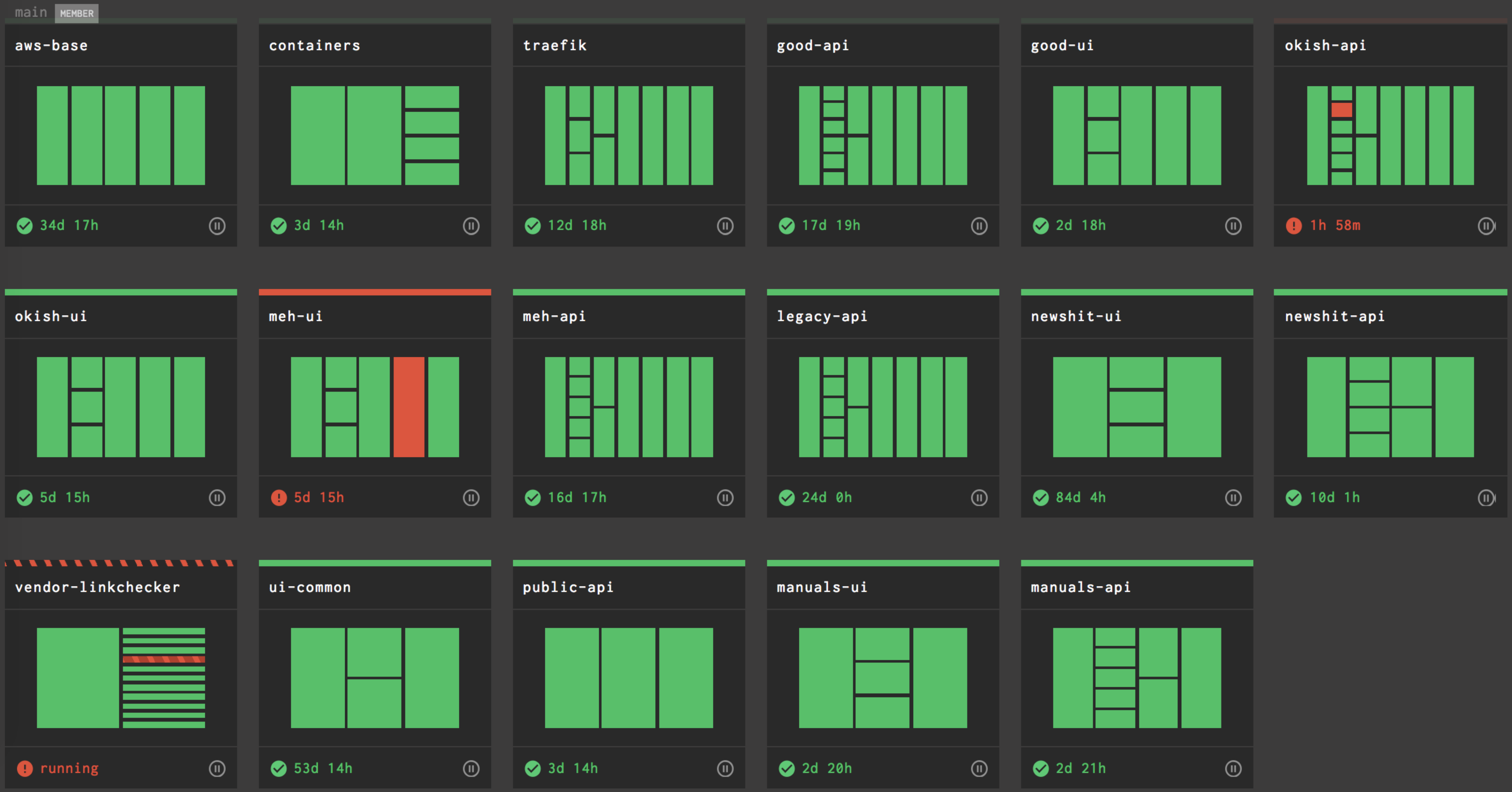
Declarative style

```
private ImmutableList<Vin> vinLists(HttpHeaders httpHeaders) {
    return Arrays.asList(
            httpHeaders.getFirst(HEADER_USER_VINLIST),
            httpHeaders.getFirst(HEADER_SECOND_USER_VINLIST),
            httpHeaders.getFirst(HEADER_USER_EMPLOYEE)
        .stream()
        .map(header -> toVinList(header))
        .flatMap(l -> l.stream())
        .distinct()
        .collect(ImmutableList.toImmutableList());
}
```

Ending point


Path to production
Devs
Code
Repo
Checks
Test env
Checks
Prod

May, 2019



main MEMBER

aws-base
✓ 34d 17h

containers
✓ 3d 14h

traefik
✓ 12d 18h

good-api
✓ 17d 19h

good-ui
✓ 2d 18h

okish-api
! 1h 58m

okish-ui
✓ 5d 15h

meh-ui
! 5d 15h

meh-api
✓ 16d 17h

legacy-api
✓ 24d 0h

newshit-ui
✓ 84d 4h

newshit-api
✓ 10d 1h

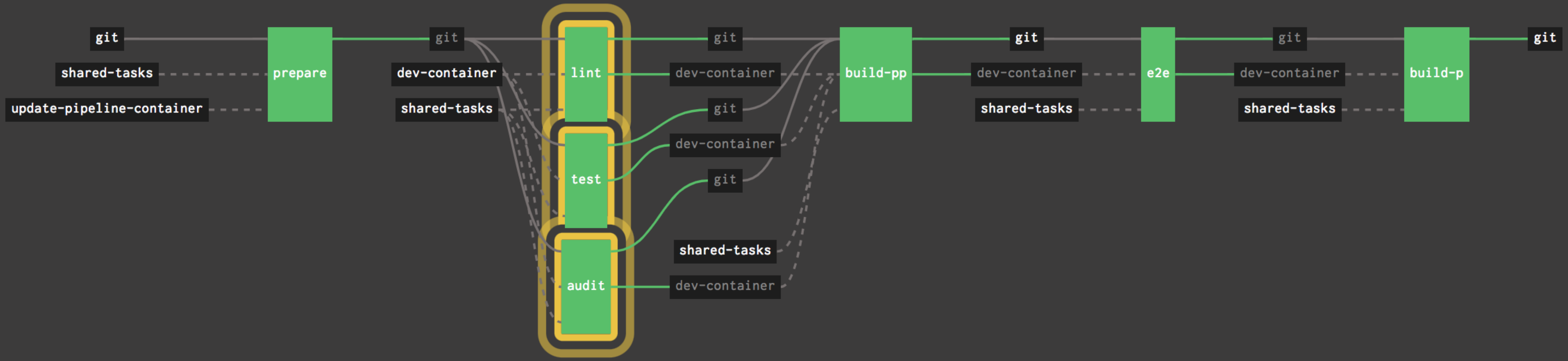
vendor-linkchecker
! running

ui-common
✓ 53d 14h

public-api
✓ 3d 14h

manuals-ui
✓ 2d 20h

manuals-api
✓ 2d 21h



git
shared-tasks
update-pipeline-container
prepare
git
dev-container
shared-tasks
lint
test
audit
git
dev-container
git
dev-container
git
shared-tasks
dev-container
build-pp
git
dev-container
shared-tasks
e2e
git
dev-container
shared-tasks
build-p
git

Deployments every 2-4 weeks

Deployments every 2-4 weeks

Multiple deployments per day

Deployments every 2-4 weeks

Manual regression testing

Multiple deployments per day

Deployments every 2-4 weeks

Manual regression testing

Multiple deployments per day

Continuous Deployment

Deployments every 2-4 weeks

Manual regression testing

Many open bugs

Multiple deployments per day

Continuous Deployment

Deployments every 2-4 weeks	Multiple deployments per day
Manual regression testing	Continuous Deployment
Many open bugs	Zero bug policy

Deployments every 2-4 weeks	Multiple deployments per day
Manual regression testing	Continuous Deployment
Many open bugs	Zero bug policy
Unpredictable cadence	

Deployments every 2-4 weeks	Multiple deployments per day
Manual regression testing	Continuous Deployment
Many open bugs	Zero bug policy
Unpredictable cadence	Fairly predictable

Deployments every 2-4 weeks	Multiple deployments per day
Manual regression testing	Continuous Deployment
Many open bugs	Zero bug policy
Unpredictable cadence	Fairly predictable
Regular delays	

Deployments every 2-4 weeks	Multiple deployments per day
Manual regression testing	Continuous Deployment
Many open bugs	Zero bug policy
Unpredictable cadence	Fairly predictable
Regular delays	Commitments reached

Your causes
Which types of causes do you care about?
Environmental agencies
Education
Children's charities

Conclusion

Continuous Delivery can have a huge impact in the performance of a team

It is not free. You have to invest to gain

It is never done. You have to keep working and improving

JOIN OUR COMMUNITY

26 years **experience**

42 offices **in 13 countries**

Thought leader **in agile software development and continuous delivery**

6000**+ thoughtworkers worldwide**

300**+ thoughtworkers in Germany**

de-recruiting@thoughtworks.com