CONTINUOUS DELIVERY

LIKE YOU KNOW NO FEAR 😱

THE RUNWAY METAPHOR

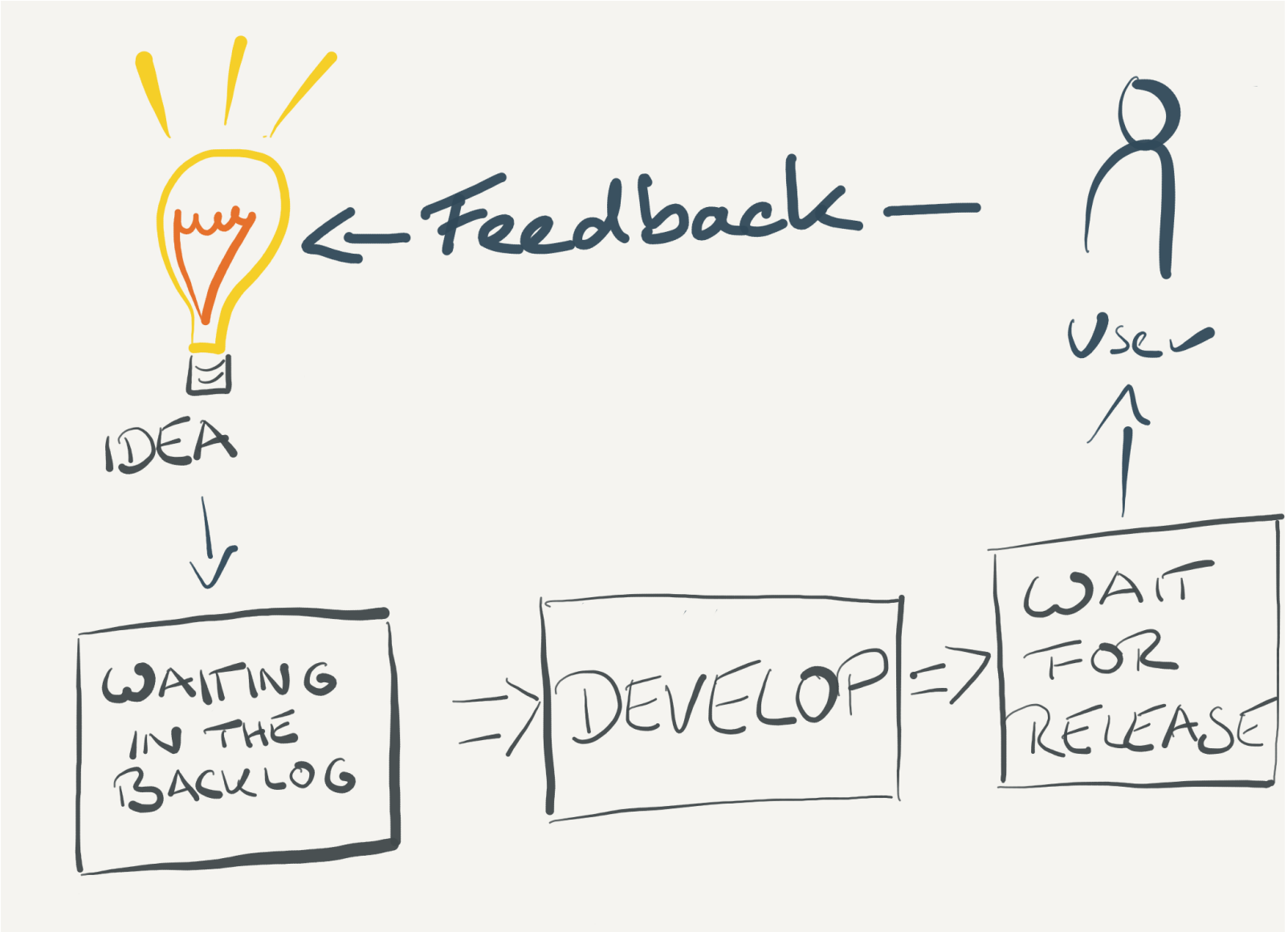
THE DOWNHILL METAPHOR

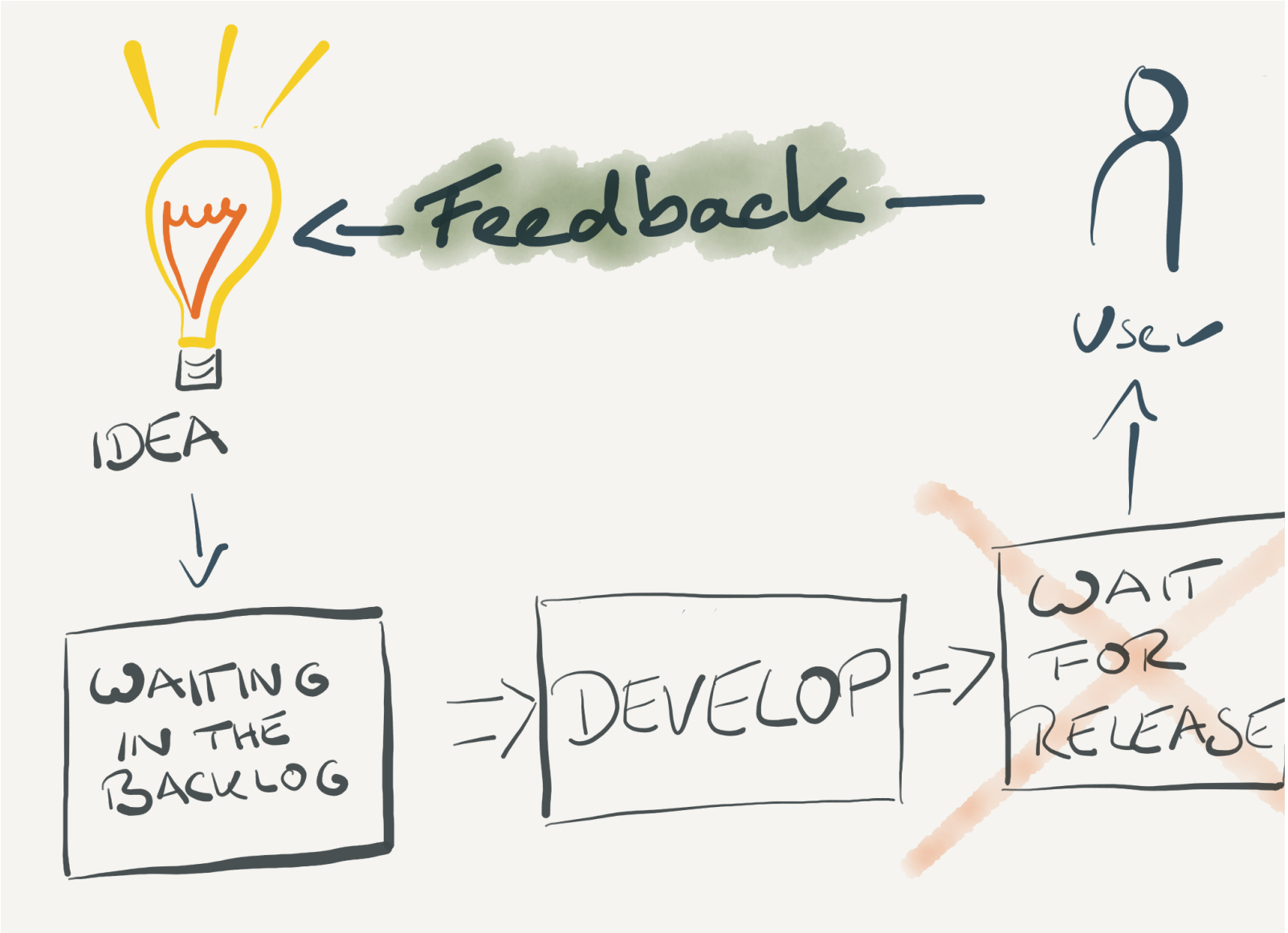
ADJUST QUICKLY IS A *VERY* IMPORTANT BUSINESS REQUIREMENT

BIG RISKS

- building the wrong product
- being too slow

It's our duty as software professionals to get ideas out there as fast as possible*


Feedback
User
IDEA
WAITING IN THE BACKLOG
DEVELOP
WAIT FOR RELEASE


Feedback
IDEA
User
WAITING IN THE BACKLOG
DEVELOP
WAIT FOR RELEASE

WHAT MEANS DONE?

WHAT MEANS DONE?

- to you

WHAT MEANS DONE?

- to you
- to your PO

WHAT MEANS DONE?

- to you
- to your PO
- to your CEO

WHAT MEANS DONE?

- to you
- to your PO
- to your CEO
- to your Customer

Done means live and available to the customer, everything else is just an excuse

Me, for the lack of a better definition

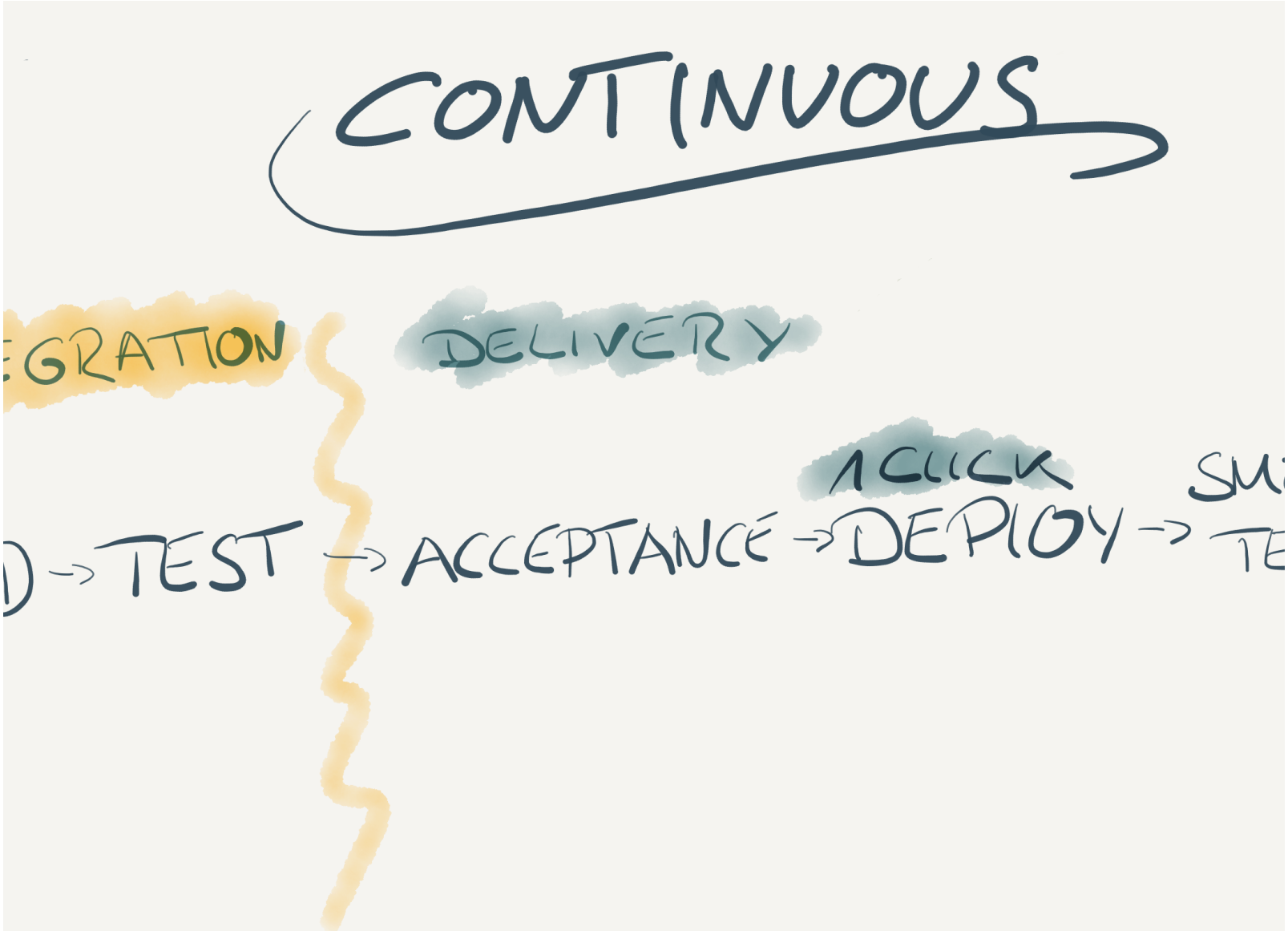
WHAT IS CONTINUOUS

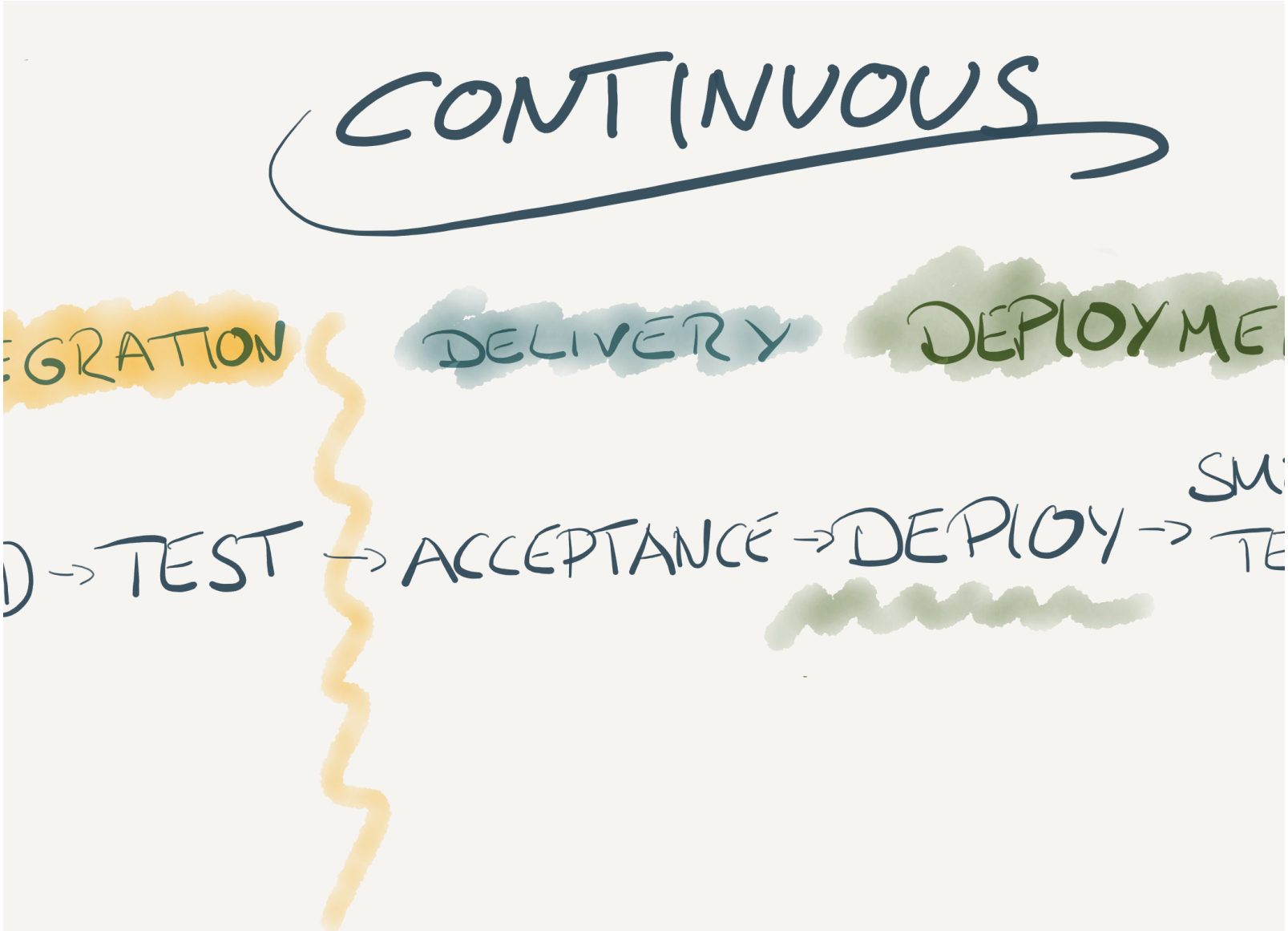
- -integration
- -delivery
- -deployment

CONTINUOUS

EGRATION

D → TEST →


CONTINUOUS
EGRATION
DELIVERY
1 CLICK
D → TEST → ACCEPTANCE → DEPLOY →
SM
TE

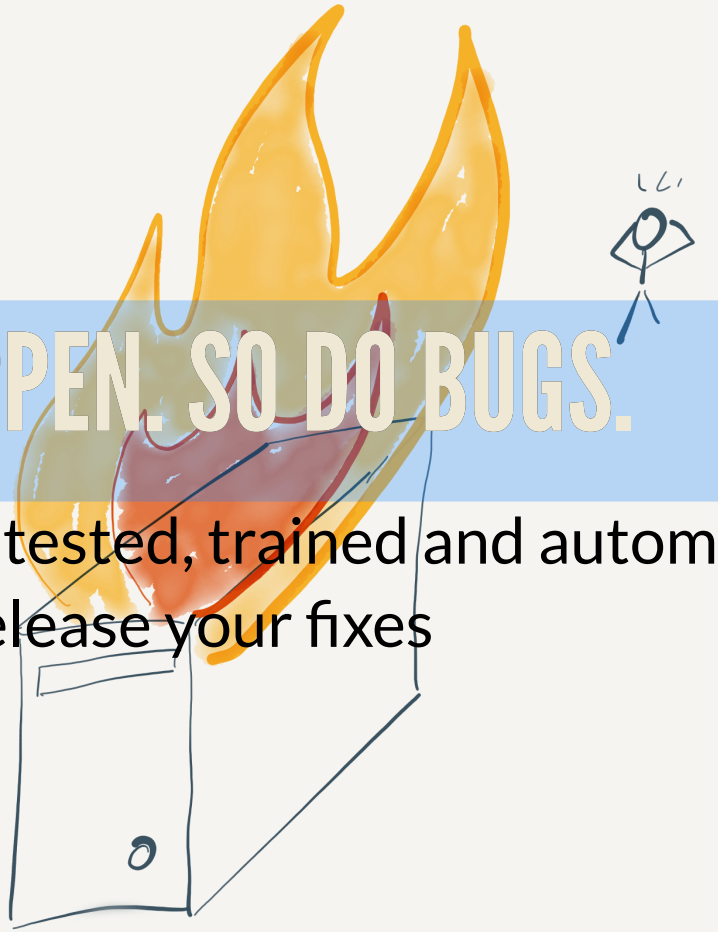

CONTINUOUS
EGRATION
DELIVERY
DEPLOYME
D -> TEST -> ACCEPTANCE -> DEPLOY -> SM
TE

WE DEPLOY ON FRIDAYS

Any not-deploying rule is a synonym for lack of tests, lack of trust or lack of feedback

WHAT TO OPTIMIZE FOR

MTBF vs. MTTR



ERRORS HAPPEN. SO DO BUGS.

That's when you need a tested, trained and automated way to release your fixes

DELIVERY PERFORMANCE

	Elite	High	Mid	Low
Freq	> 1/h	1/h - 1/d	1/w - 1/m	1/w - 1/m
Lead Time	< 1h	1d - 1w	1w - 1m	1m - 6m
MTTR	<1h	<1d	<1d	1w - 1m
CFR	0-15%	0-15%	0-15%	46-60%

TIME SPENT

	Elite	High	Mid	Low
new work	50%	50%	40%	30%
rework	19,5%	20%	20%	20%
security	5%	5%	5%	10%
defects	10%	10%	10%	20%
support	5%	10%	10%	15%

[DORA State of DevOps Report 2018 p32]

DEMO

THE MATMATCH PATH TO CD

THE MATMATCH PATH TO CD

1. Fix deploy automation (1 command)

THE MATMATCH PATH TO CD

1. Fix deploy automation (1 command)
2. Fix Testing

THE MATMATCH PATH TO CD

1. Fix deploy automation (1 command)
2. Fix Testing
3. Fix Review Process

THE MATMATCH PATH TO CD

1. Fix deploy automation (1 command)
2. Fix Testing
3. Fix Review Process
4. Fix Observability

THE MATMATCH PATH TO CD

1. Fix deploy automation (1 command)
2. Fix Testing
3. Fix Review Process
4. Fix Observability
5. Fix Release Process

THE MATMATCH PATH TO CD

1. Fix deploy automation (1 command)
2. Fix Testing
3. Fix Review Process
4. Fix Observability
5. Fix Release Process
6. Fix Fear

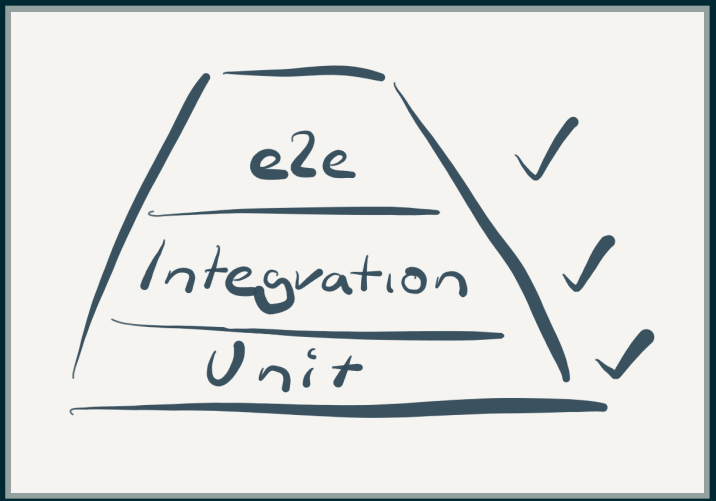
GETTING THERE

```
make deploy production
```

GETTING THERE

FIX TESTING

separate our tests into different build steps for quicker feedback



GETTING THERE

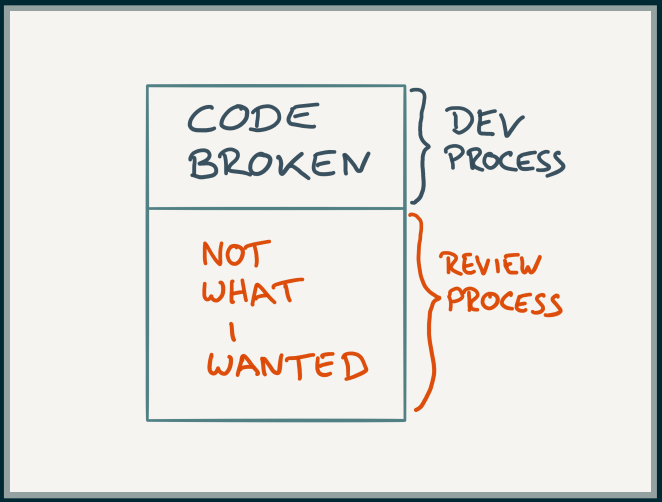
FIX TESTING

speed up the build -> build on big ec2 spot instances

(we went from TravisCI over AWS CodeBuild to Gitlab Runners)

GETTING THERE

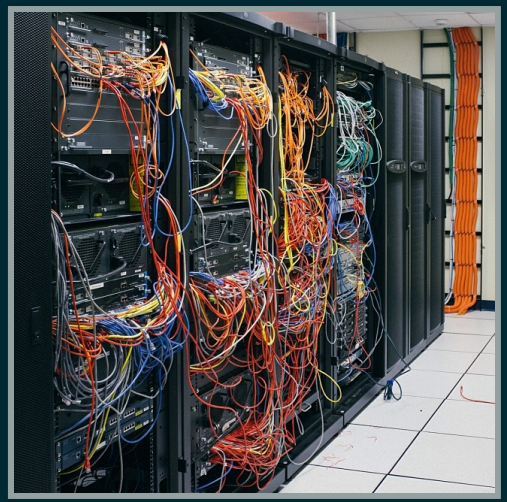
FIX THE REVIEW PROCESS



GETTING THERE

FIX OUR ENVIRONMENT

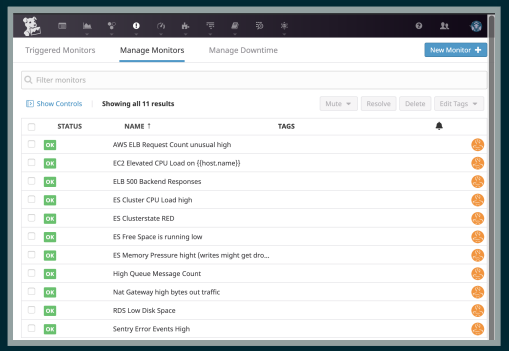
Dependable releases need dependable infrastructure

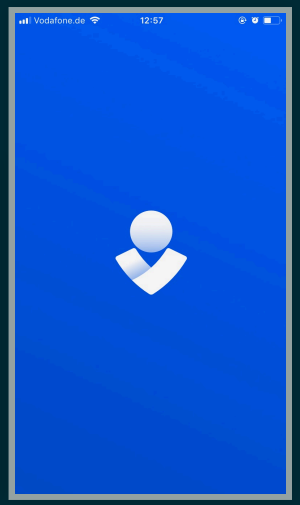


GETTING THERE

OBSERVABILITY OF THE SYSTEM

get good metrics and feed them into good tools





GETTING THERE

COPS OF THE DAY

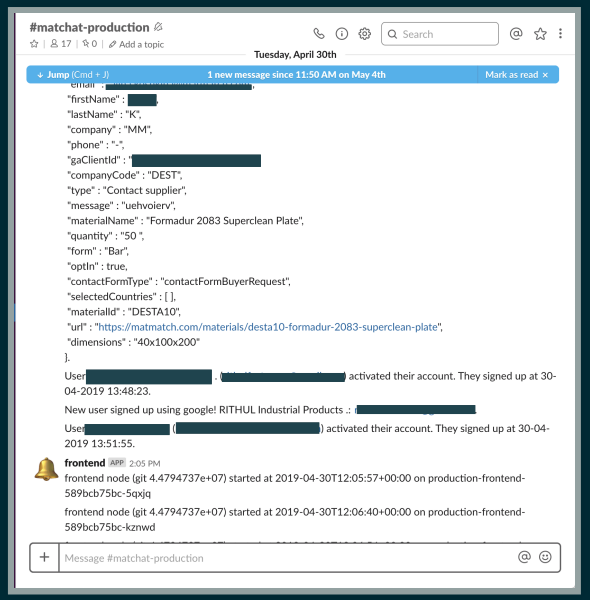
Increase operational knowledge in the Team:

random pair of developers run the "morning checklist"

GETTING THERE

CHATOPS

be involved in the system behavior - convert important business events to chat messages

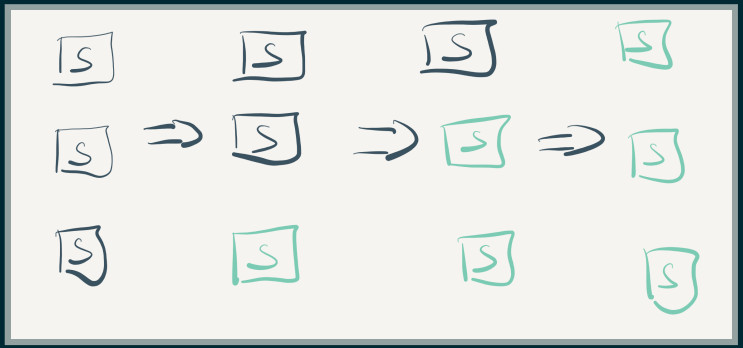


GETTING THERE

FIX RELEASE PROCESS

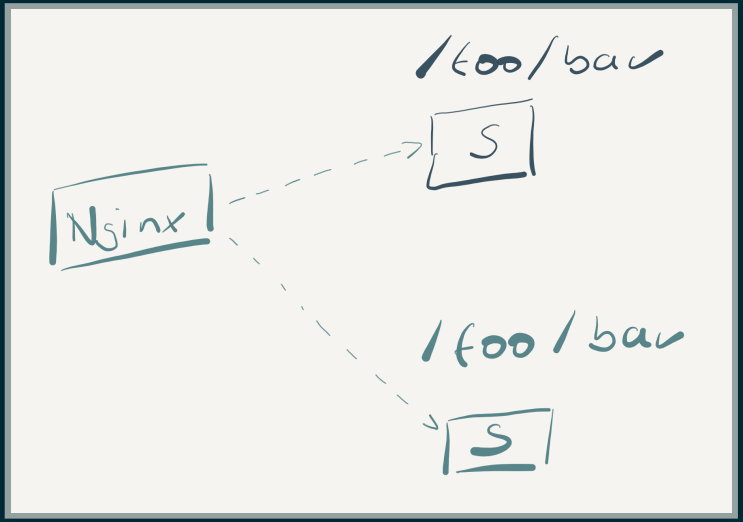
GETTING THERE

ROLLING RELEASES



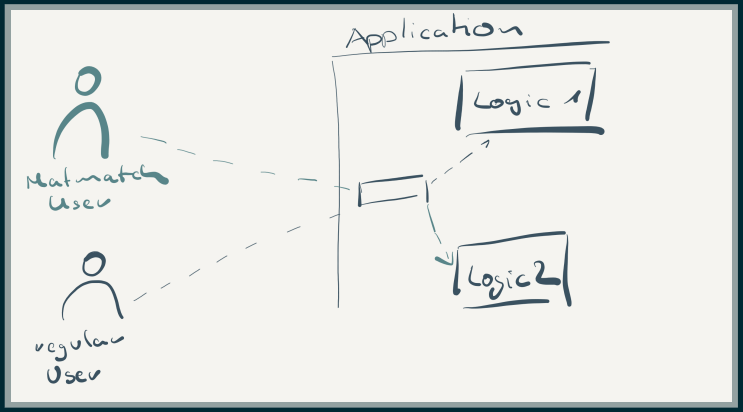
GETTING THERE

DARK RELEASES



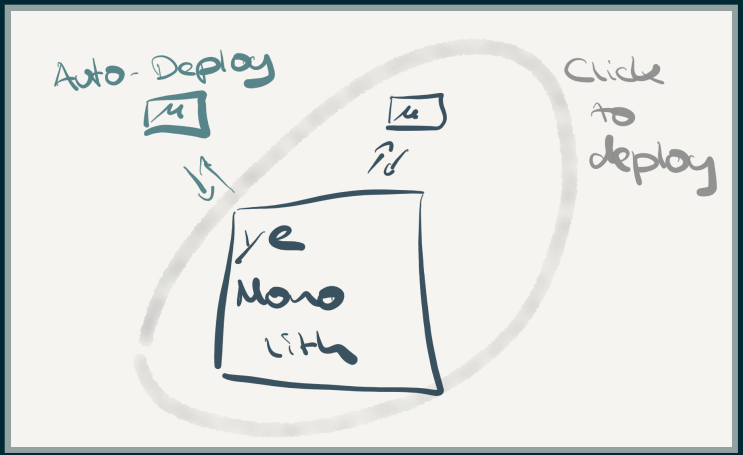
GETTING THERE

RIGHTS BASED RELEASES



GETTING THERE

LIMIT FEAR: SMALL RELEASES



GETTING THERE

LIMIT FEAR: MAKE IT BORING

GETTING THERE

LIMIT FEAR: MAKE IT BORING

1. Automate everything until the "Deploy" Button (Continous Delivery)

GETTING THERE

LIMIT FEAR: MAKE IT BORING

1. Automate everything until the "Deploy" Button (Continous Delivery)
2. Make pushing the Button a boring non-event

GETTING THERE

LIMIT FEAR: MAKE IT BORING

1. Automate everything until the "Deploy" Button (Continous Delivery)
2. Make pushing the Button a boring non-event
3. Remove the Button, no one cares

ISSUES WE FACED

ISSUES WE FACED

- K8S: Health and Readiness Endpoints

ISSUES WE FACED

- K8S: Health and Readiness Endpoints
- K8S: Pod Limits

ISSUES WE FACED

- K8S: Health and Readiness Endpoints
- K8S: Pod Limits
- K8S Overload - Rebuild everything from scratch

ISSUES WE FACED

- K8S: Health and Readiness Endpoints
- K8S: Pod Limits
- K8S Overload - Rebuild everything from scratch
- Dependencies between Services (Stateful Set by Accident)

ISSUES WE FACED

- K8S: Health and Readiness Endpoints
- K8S: Pod Limits
- K8S Overload - Rebuild everything from scratch
- Dependencies between Services (Stateful Set by Accident)

=> Architectural and Ops Problems bite you twice as fast

ISSUES WE FACED

- K8S: Health and Readiness Endpoints
- K8S: Pod Limits
- K8S Overload - Rebuild everything from scratch
- Dependencies between Services (Stateful Set by Accident)

=> Architectural and Ops Problems bite you twice as fast

=> But you notice them before it hits production

SUMMARY

- We are faster than before
- We have less defects and issues than before
- We have a ❤️ for production

5,25 Humans => up to 15 Releases / Day 🔥

KEY TAKEAWAYS

KEY TAKEAWAYS

- CD pays off big time

KEY TAKEAWAYS

- CD pays off big time
- Slow and steady wins the race to get there

KEY TAKEAWAYS

- CD pays off big time
- Slow and steady wins the race to get there
- It's not just tech, it's also emotions

KEY TAKEAWAYS

- CD pays off big time
- Slow and steady wins the race to get there
- It's not just tech, it's also emotions
- Don't play too much with your K8S Prod Cluster

KEY TAKEAWAYS

- CD pays off big time
- Slow and steady wins the race to get there
- It's not just tech, it's also emotions
- Don't play too much with your K8S Prod Cluster
- Most things you have to fix for CD are things help you a lot in general

SUGGESTED READINGS

- Continuous Delivery, Jez Humble
- Accelerate, Nicole Forsgren Phd
- Release It, Michael Nygard
- The Phoenix Project, Gene Kim

QUESTIONS?

My twitter is @chrisuhl

Examples, slides and everything else on github:
https://github.com/christianuhlcc/cd-like-you-mean-it