CONTINUOUS DELIVERY

LIKE YOU KNOW NO FEAR 😱

I'm Chris

I work for a munich based startup called Matmatch

Matmatch

We're a development team of 5 (and me) trying to change the world of materials

THE RUNWAY
METAPHOR
24

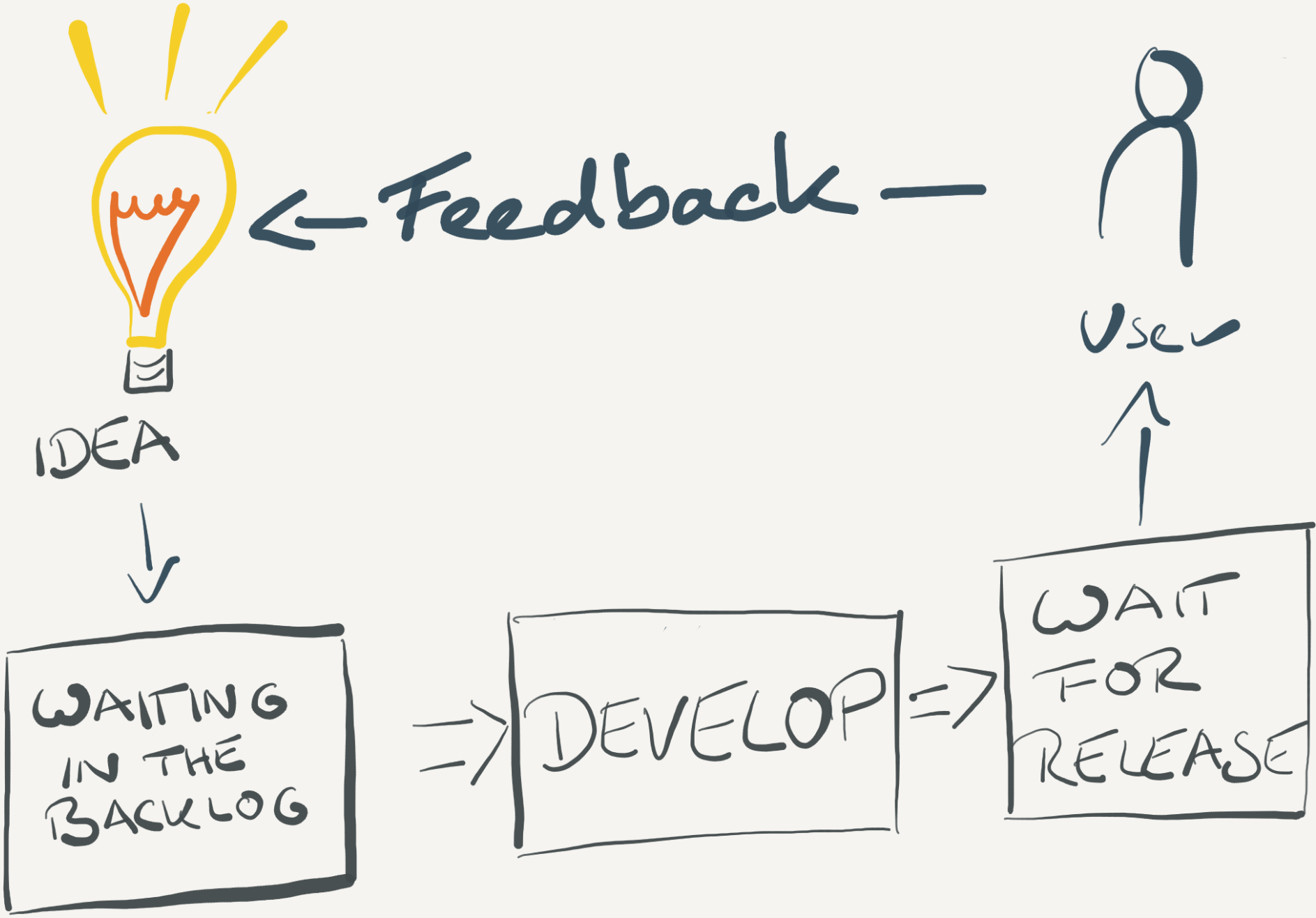
THE DOWNHILL METAPHOR

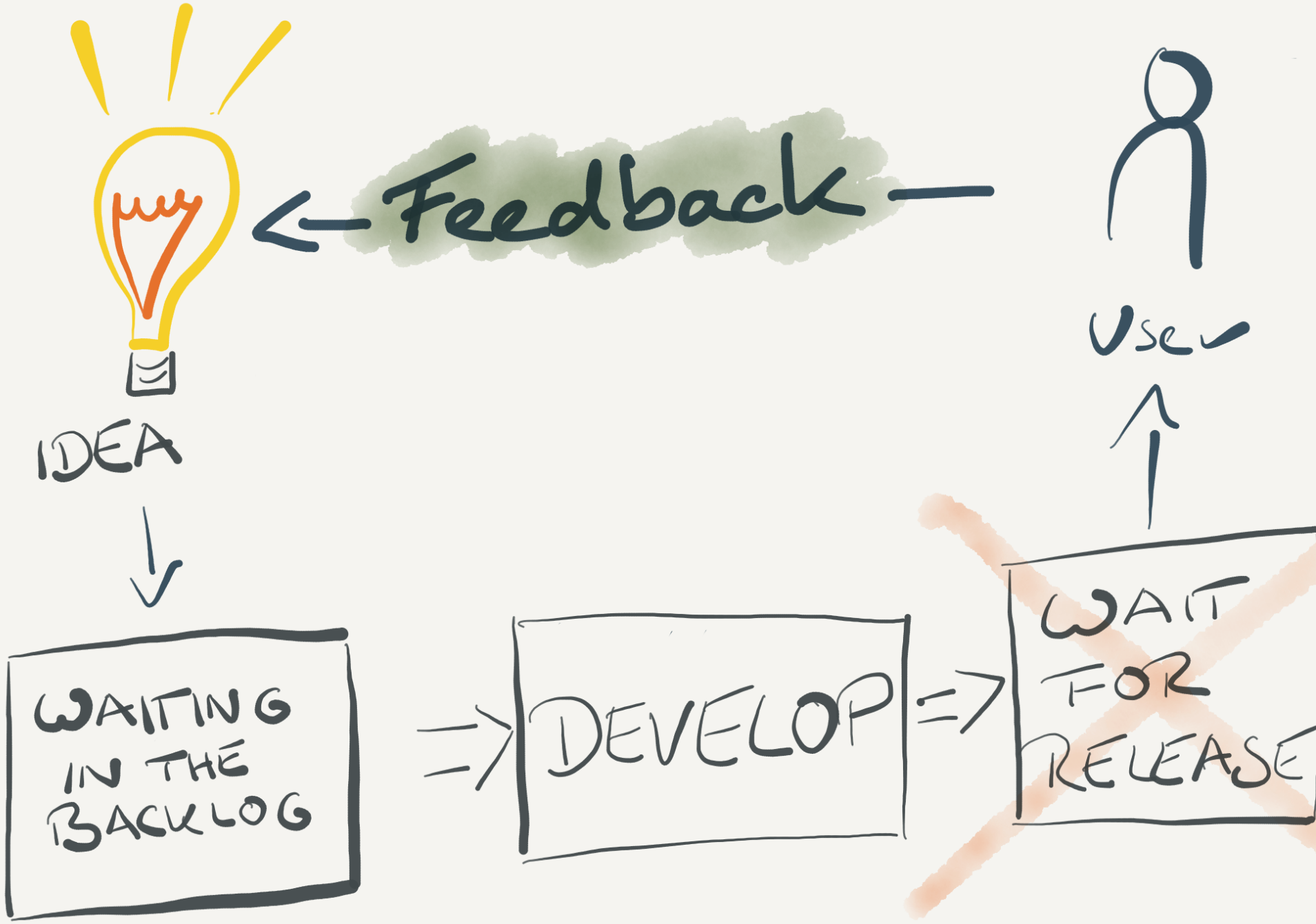
ADJUST QUICKLY IS A *VERY* IMPORTANT

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
IDEA
User
WAITING IN THE BACKLOG
DEVELOP
WAIT FOR RELEASE


Feedback
IDEA
User
WAITING IN THE BACKLOG
DEVELOP
WAIT FOR RELEASE

WHAT DOES DONE MEAN?

WHAT DOES DONE MEAN?

- to you

WHAT DOES DONE MEAN?

- to you
- to your PO

WHAT DOES DONE MEAN?

- to you
- to your PO
- to your CEO

WHAT DOES DONE MEAN?

- 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

SM

) → TEST → ACCEPTANCE → DEPLOY → TE

CONTINUOUS

EGRATION DELIVERY DEPLOYME

D → TEST → ACCEPTANCE → DEPLOY → SM
TE

WE DEPLOY ON FRIDAYS

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

WHAT TO OPTIMIZE FOR

MTBF VS. MTTR

Mean Time Between Failure *vs* Mean Time To Recovery

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

please go to https://demo.matmatch.com

THE MATMATCH PATH TO CD

THE MATMATCH PATH TO CD

1. Fix Deployment Automation

THE MATMATCH PATH TO CD

1. Fix Deployment Automation
2. Fix Testing

THE MATMATCH PATH TO CD

1. Fix Deployment Automation
2. Fix Testing
3. Fix Review Process

THE MATMATCH PATH TO CD

1. Fix Deployment Automation
2. Fix Testing
3. Fix Review Process
4. Fix Observability

THE MATMATCH PATH TO CD

1. Fix Deployment Automation
2. Fix Testing
3. Fix Review Process
4. Fix Observability
5. Fix Release Process

THE MATMATCH PATH TO CD

1. Fix Deployment Automation
2. Fix Testing
3. Fix Review Process
4. Fix Observability
5. Fix Release Process
6. Fix Fear

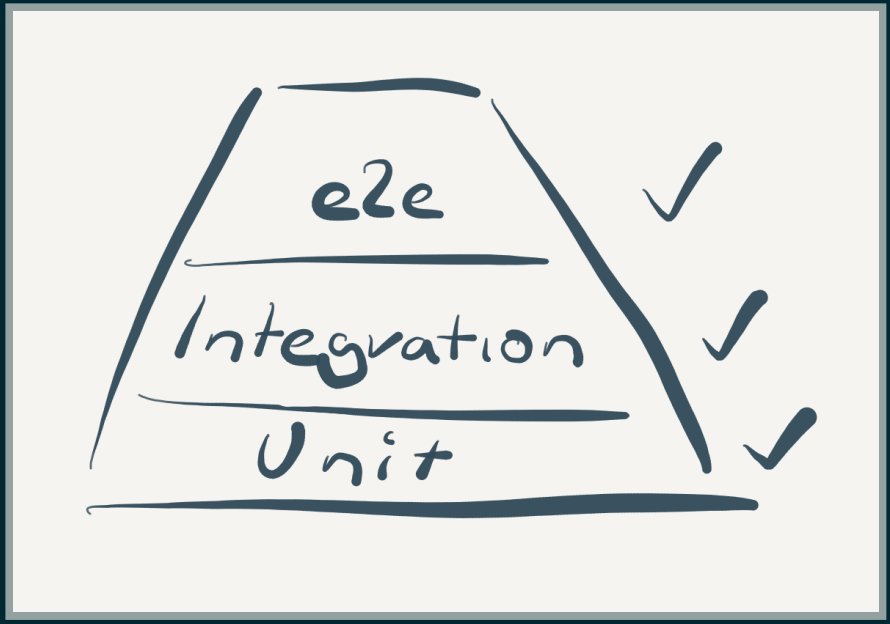
GETTING THERE

```
make deploy production
```

GETTING THERE

FIX TESTING

Separate our tests into different build steps for quicker feedback



GETTING THERE

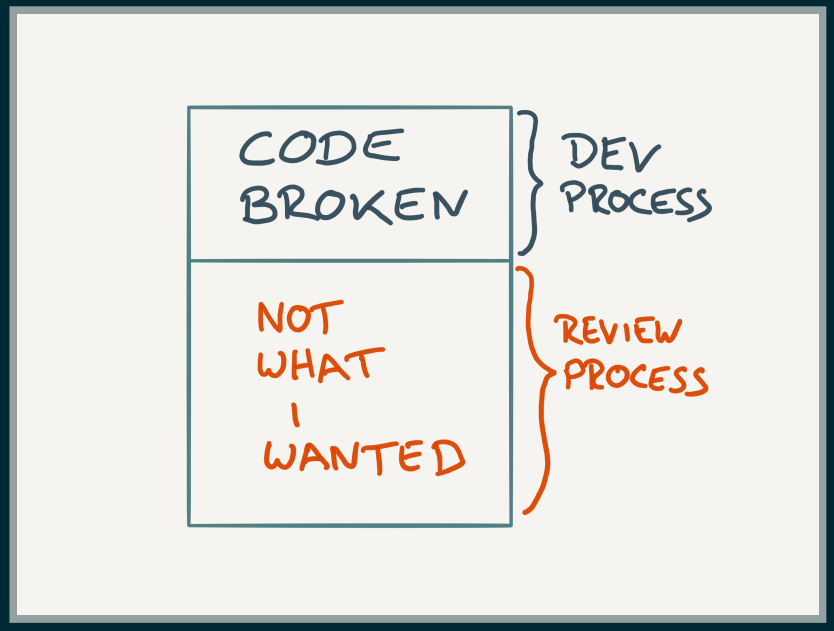
FIX TESTING

Speed up the build -> Build on big ec2 spot instances

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

GETTING THERE

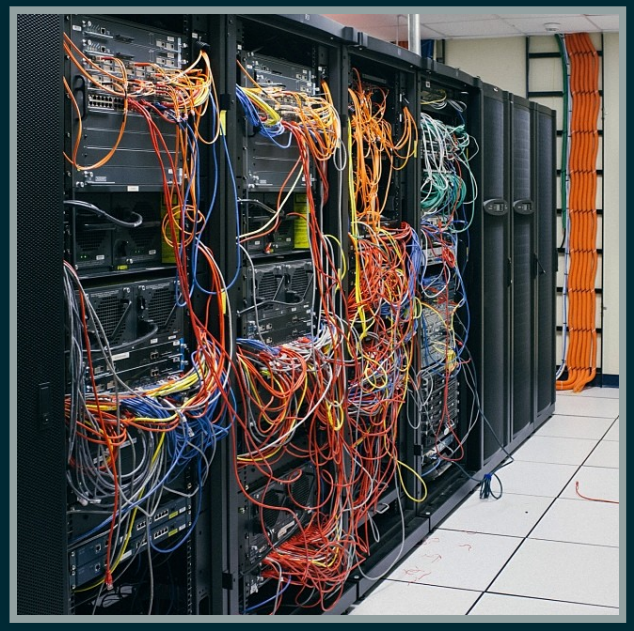
FIX THE REVIEW PROCESS



GETTING THERE

FIX OUR ENVIRONMENT

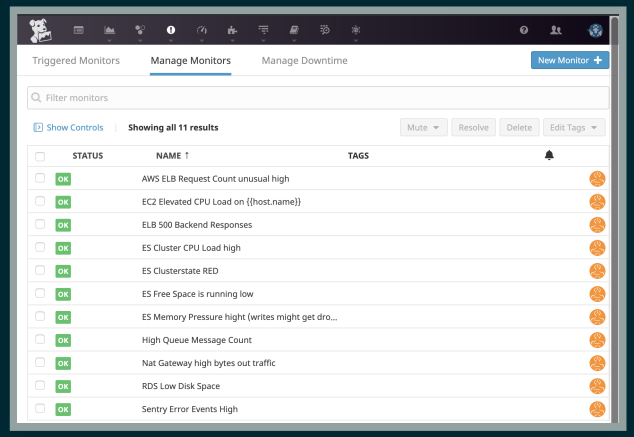
Dependable releases need dependable infrastructure

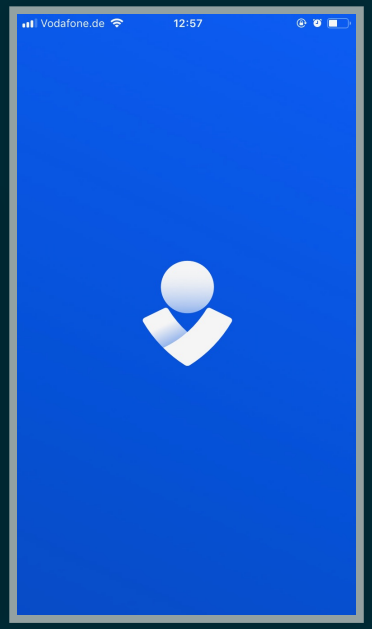


GETTING THERE

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

#matchat-production

17 | 0 | Add a topic

Search

Tuesday, April 30th

Jump (Cmd + J) 1 new message since 11:50 AM on May 4th Mark as read ×

"email" : [redacted],
"firstName" : [redacted],
"lastName" : "K",
"company" : "MM",
"phone" : "-",
"gaClientId" : "[redacted]
"companyCode" : "DEST",
"type" : "Contact supplier",
"message" : "uehvoierv",
"materialName" : "Formadur 2083 Superclean Plate",
"quantity" : "50 ",
"form" : "Bar",
"optIn" : true,
"contactFormType" : "contactFormBuyerRequest",
"selectedCountries" : [ ],
"materialId" : "DESTA10",
"url" : "https://matmatch.com/materials/desta10-formadur-2083-superclean-plate",
"dimensions" : "40x100x200"
}.

User [redacted] . ([redacted]) activated their account. They signed up at 30-04-2019 13:48:23.

New user signed up using google! RITHUL Industrial Products .: [redacted]

User [redacted] ([redacted]) activated their account. They signed up at 30-04-2019 13:51:55.

frontend APP 2:05 PM

frontend node (git 4.4794737e+07) started at 2019-04-30T12:05:57+00:00 on production-frontend-589bcb75bc-5qxjq

frontend node (git 4.4794737e+07) started at 2019-04-30T12:06:40+00:00 on production-frontend-589bcb75bc-kznwd

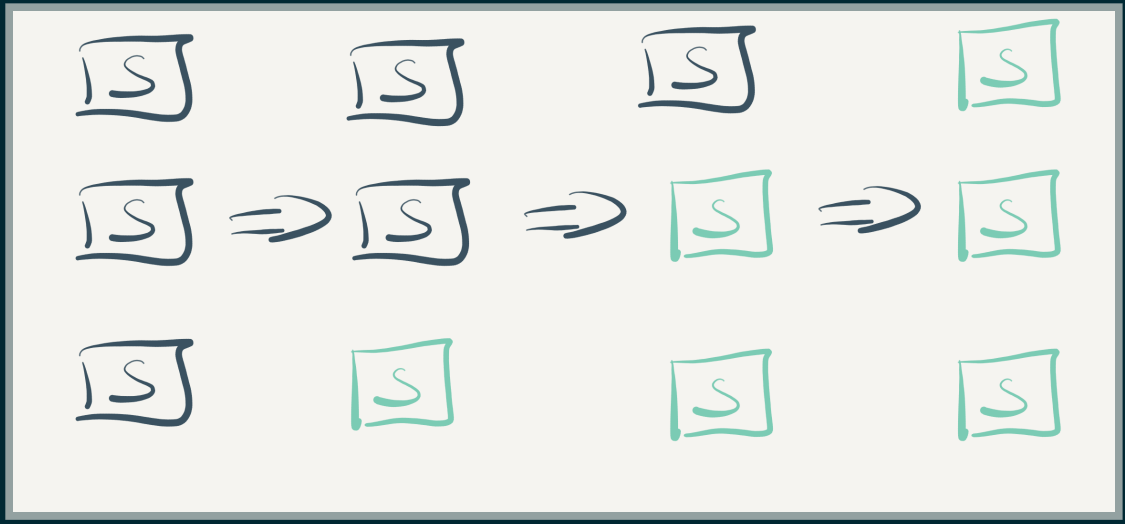
Message #matchat-production

GETTING THERE

FIX RELEASE PROCESS

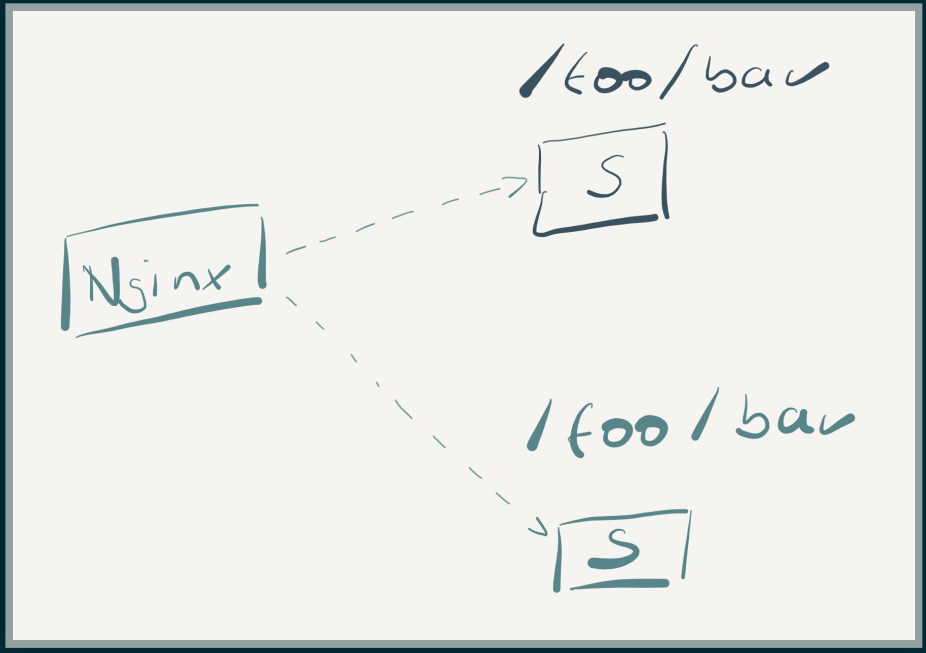
GETTING THERE

ROLLING RELEASES



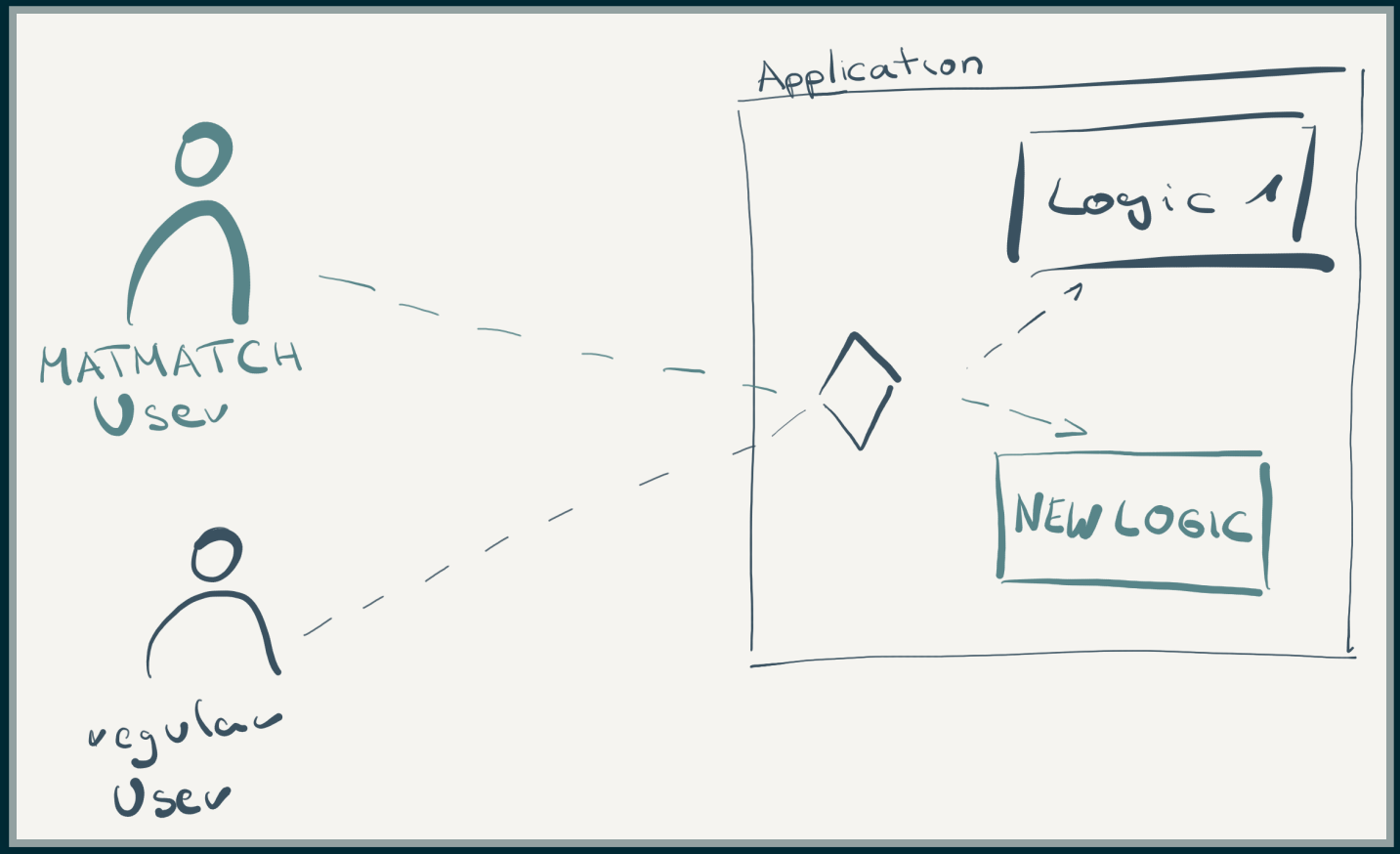
GETTING THERE

DARK RELEASES



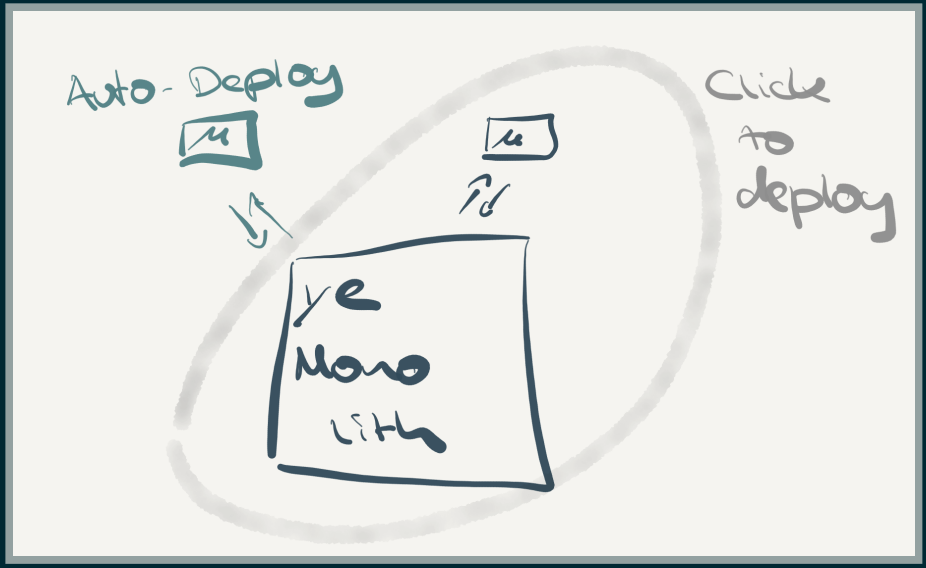
GETTING THERE

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 (Continuous Delivery)

GETTING THERE

LIMIT FEAR: MAKE IT BORING

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

GETTING THERE

LIMIT FEAR: MAKE IT BORING

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

ISSUES WE FACED

ISSUES WE FACED

- Learning Helm was more effort than we thought

ISSUES WE FACED

- Learning Helm was more effort than we thought
- K8S: Health and readiness endpoints

ISSUES WE FACED

- Learning Helm was more effort than we thought
- K8S: Health and readiness endpoints
- K8S: Pod limits

ISSUES WE FACED

- Learning Helm was more effort than we thought
- K8S: Health and readiness endpoints
- K8S: Pod limits
- K8S overload: Rebuild everything from scratch

ISSUES WE FACED

- Learning Helm was more effort than we thought
- K8S: Health and readiness endpoints
- K8S: Pod limits
- K8S overload: Rebuild everything from scratch
- There are no multi-level-wildcard certificates

ISSUES WE FACED

- Learning Helm was more effort than we thought
- K8S: Health and readiness endpoints
- K8S: Pod limits
- K8S overload: Rebuild everything from scratch
- There are no multi-level-wildcard certificates
- Dependencies between services (stateful set by accident)

ISSUES WE FACED

- Learning Helm was more effort than we thought
- K8S: Health and readiness endpoints
- K8S: Pod limits
- K8S overload: Rebuild everything from scratch
- There are no multi-level-wildcard certificates
- Dependencies between services (stateful set by accident)
- Fixing logic in a lot of places (gitlab-ci, helm charts)

ISSUES WE FACED (SUMMARY)

ISSUES WE FACED (SUMMARY)

- Architectural and ops problems bite you twice as fast

ISSUES WE FACED (SUMMARY)

- Architectural and ops problems bite you twice as fast
- But you notice them before they hit production

ISSUES WE FACED (SUMMARY)

- Architectural and ops problems bite you twice as fast
- But you notice them before they hit production
- The rest just needs to be worked off piece by piece

ONGOING ISSUES

- Duplicated code in gitlab-ci.yml files
- Duplicated code in helm charts
- Deployments fail to clean up every once in a blue moon

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

- Continuous Deployment pays off big time

KEY TAKEAWAYS

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

KEY TAKEAWAYS

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

KEY TAKEAWAYS

- Continuous Deployment 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

- Continuous Deployment 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 that 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

w