A Whole Team Approach to Testing in Continuous Delivery

XA 2020

Lisa Crispin and Areti Panou

About us

Lisa Crispin
Lisa@agiletester.ca
Helping to build observability at
outsystems

Areti Panou
areti.panou@gmail.com
Product Owner of internal tools at
SAP

@lisacrispin

@UnremarkableQA

Learning intentions

- Get your team engaged to deploy frequently, with confidence
- Learn the language of continuous delivery
- Exercises you can do with your team to shorten feedback loops
- Ways to build a quality culture
- We’ll talk and demonstrate. Then live Q&A!

We’ve encountered a few obstacles…

We’ll each share some ideas & materials to get your team talking

Write down your questions as we go!

Live Q&A follows!

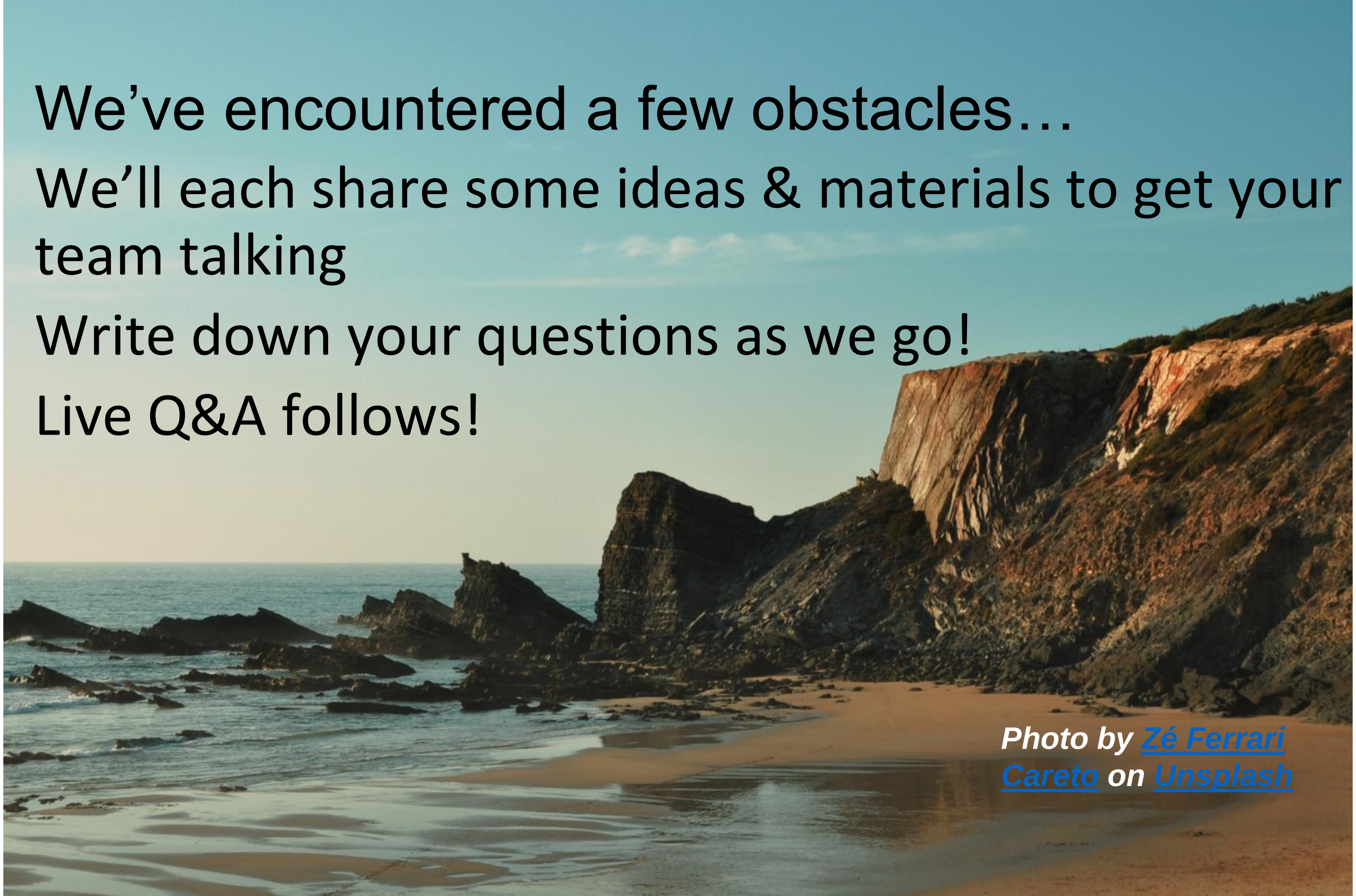
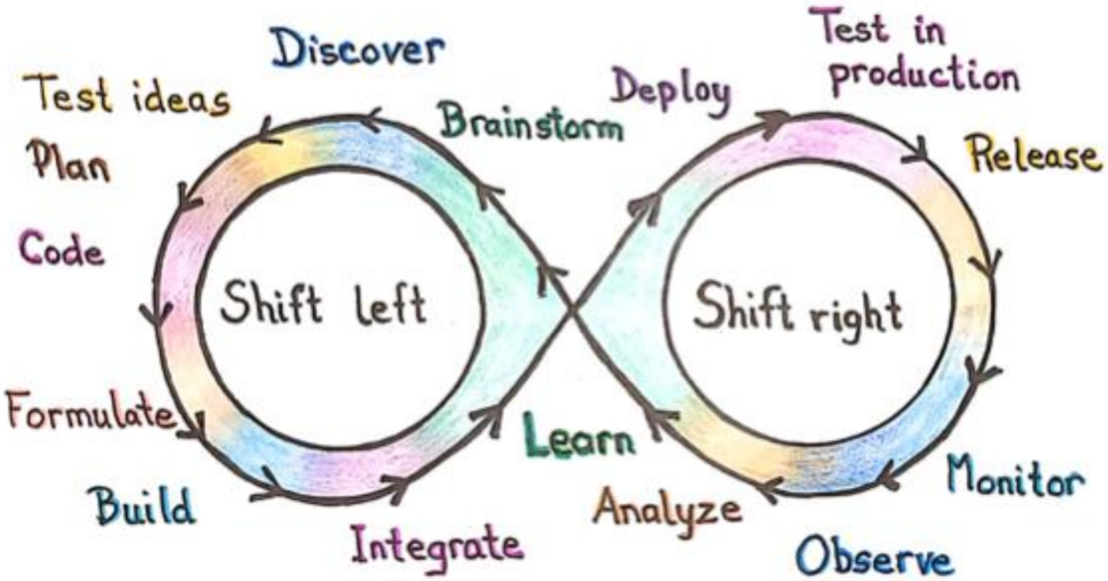


Photo by Zé Ferrari Careto on Unsplash

Building quality in

Whole team ownership of quality



Graphic by Janet Gregory, inspired by Dan Ashby

@lisacrispin

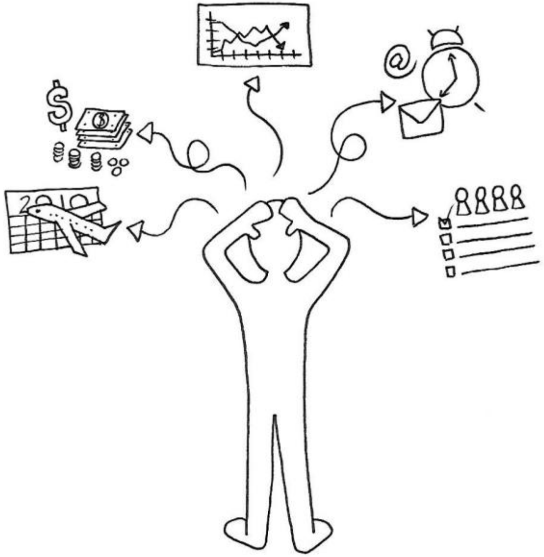
@UnremarkableQA

Whole team ownership of quality

- Share common goals
- Build relationships across roles, teams
- Continuously learn and improve
- Take advantage of multiple skill sets, perspectives
- Continuous, holistic testing

Guidelines for continual improvement

- Small slices, learning releases
- Iterative improvement
- Build towards your goal



Prioritize!

- How does it align with the business?
- Is it valuable to the customer?
- Are the stories small and testable?

Note: see resources slide for more on this

Continuous improvement...

Let's look at some tools, techniques and frameworks that can help

Infrastructure

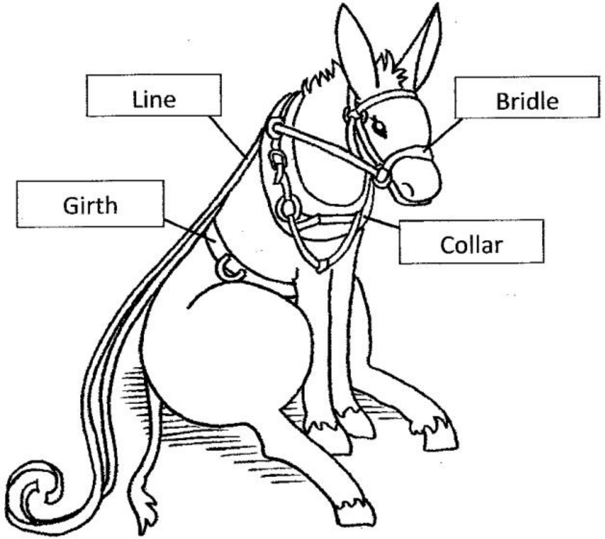
…… for fast feedback and frequent delivery

DevOps isn’t only about infrastructure

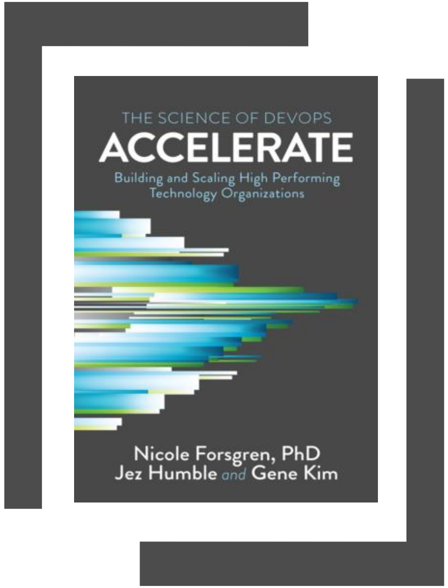
- Collaboration
- Continuous improvement
- Continuous learning
- Build quality in
- Small batches
- Everyone is responsible

© Libby VanderPloeg

Let’s agree on some common terminology



Metrics for success – per State of DevOps Survey

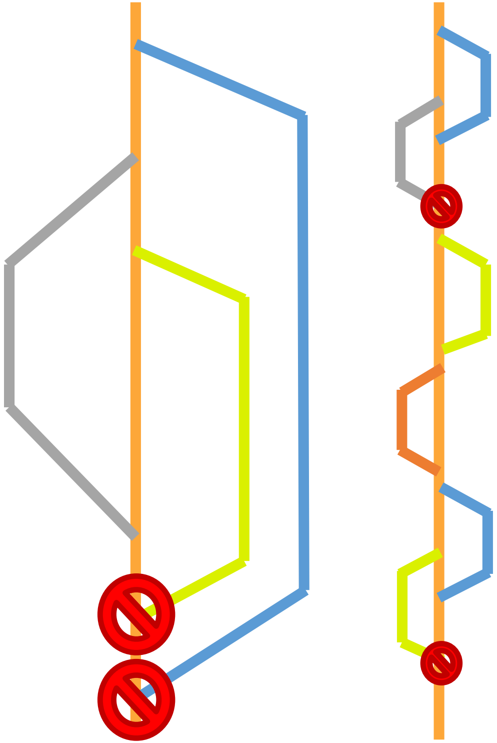


- Lead time for changes
  - Re-work slows us down
  - Shared understanding speeds us up
- Change failure rate
- Time to restore service (MTTR)
- Deployment frequency

Continuous Integration (CI)

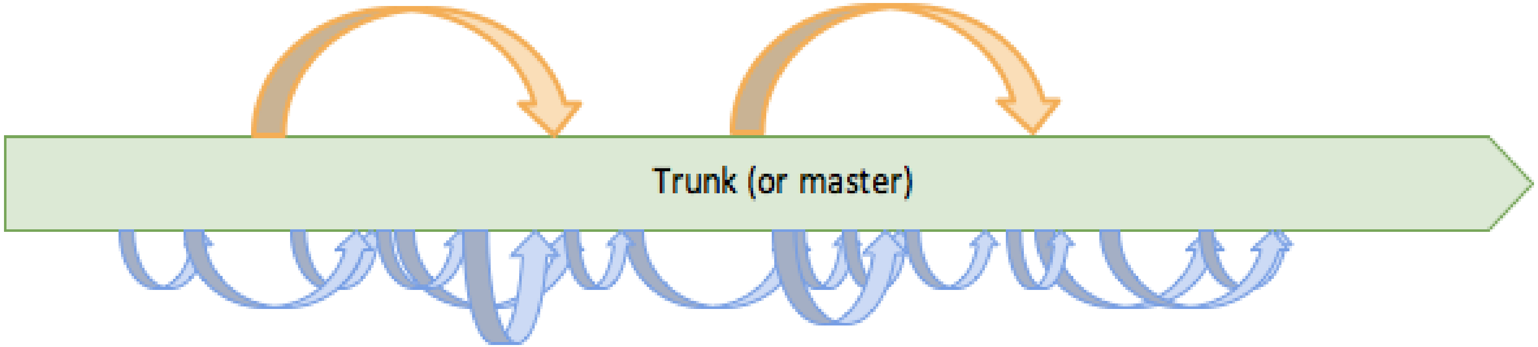
Multiple times per day:

- Integrate code into a shared repository
- All code is integrated into the same branch



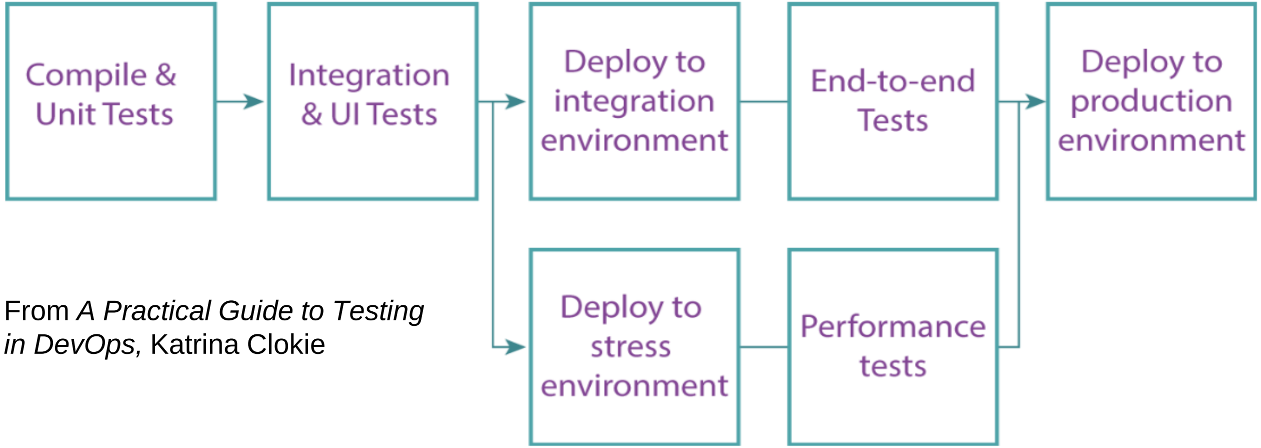
Validating each change

- Typically the start of a pipeline
- Each check-in can be verified by an automated build with automated regression tests



Deployment pipeline

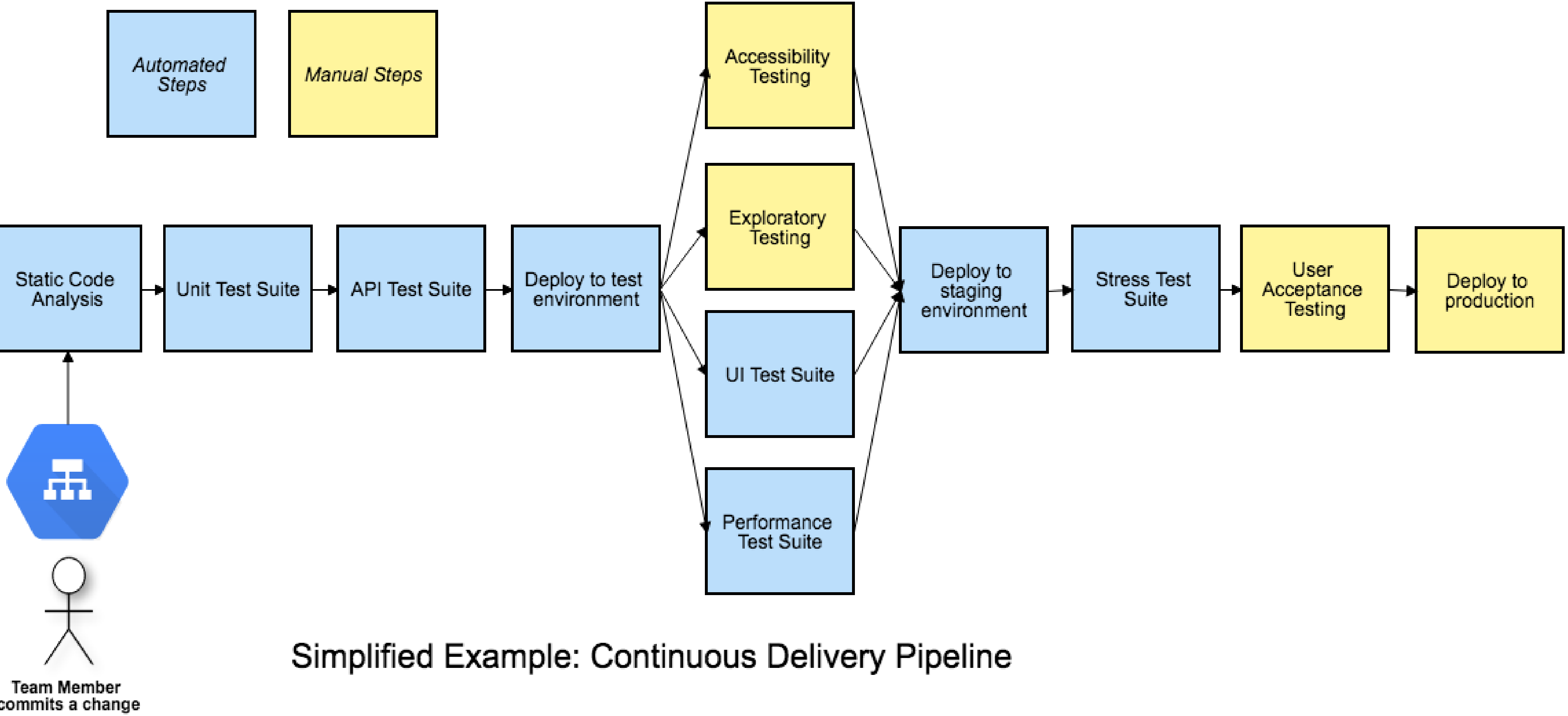
- Break the build into stages to speed up feedback
- Early stages can find most problems -> faster feedback
- Later stages probe more thoroughly
- Automated deployment pipelines are central to continuous delivery



Compile & Unit Tests
Integration & UI Tests
Deploy to integration environment
End-to-end Tests
Deploy to production environment
Deploy to stress environment
Performance tests

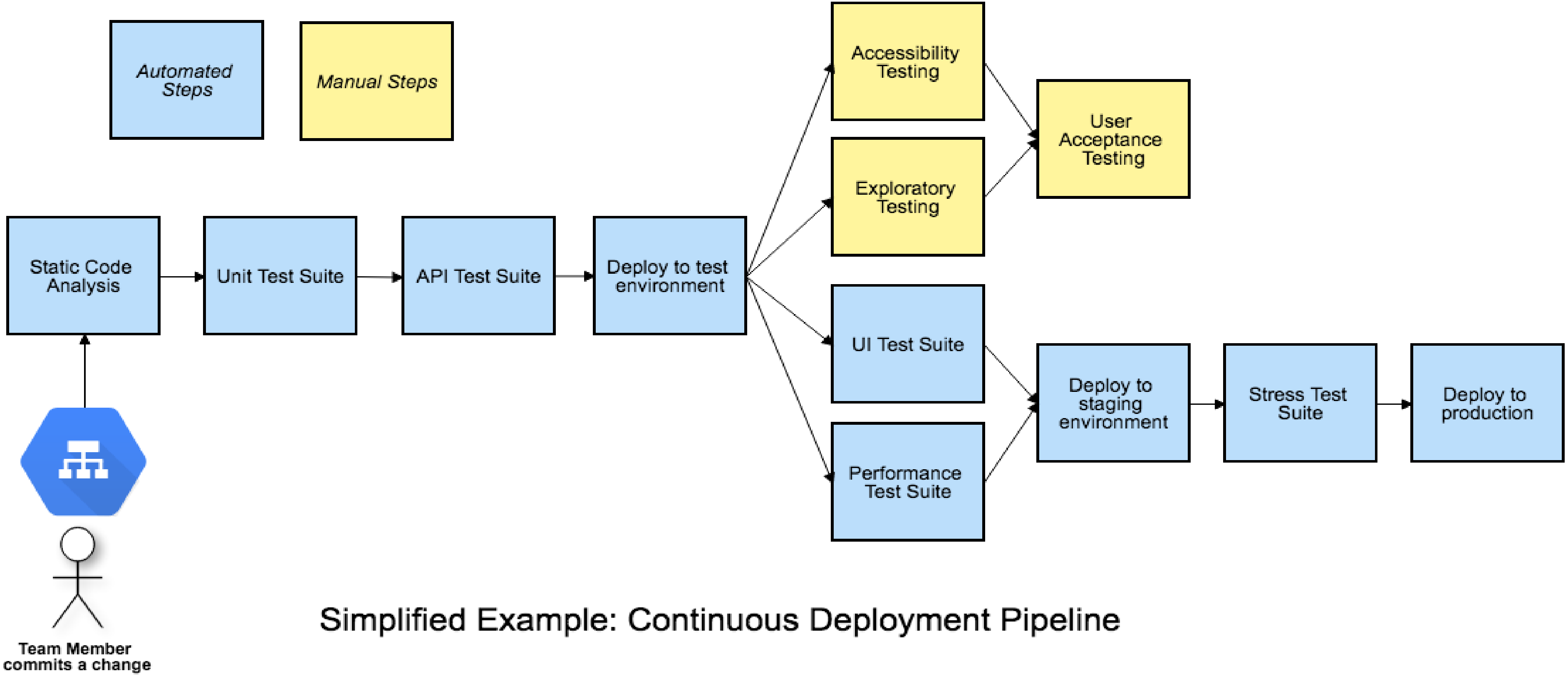
From *A Practical Guide to Testing in DevOps*, Katrina Clokie

Continuous Delivery (CD)



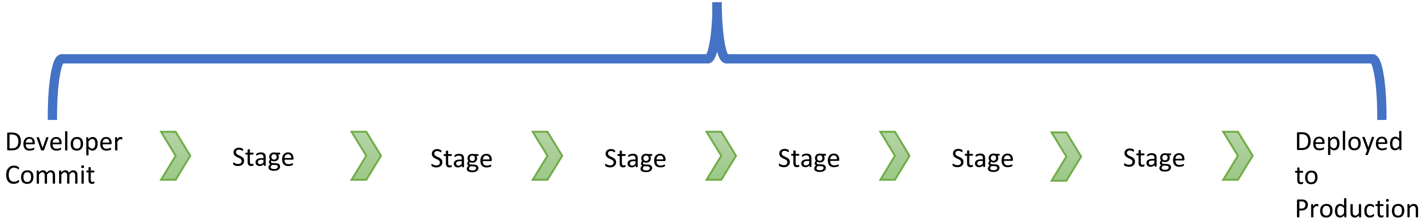
Simplified Example: Continuous Delivery Pipeline

Continuous Deployment (also CD)



Simplified Example: Continuous Deployment Pipeline

Deployment Pipeline



Developer Commit

Stage

Stage

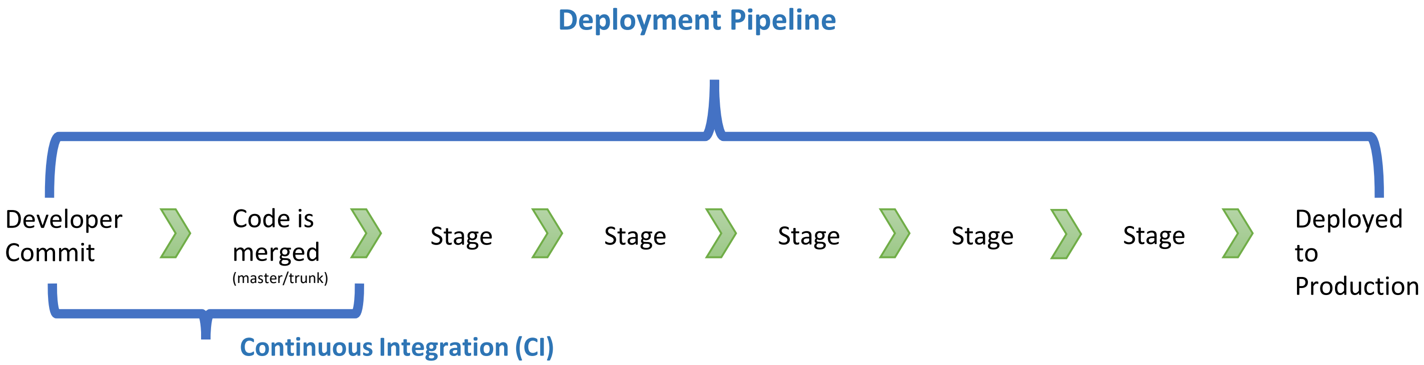
Stage

Stage

Stage

Stage

Deployed to Production



Deployment Pipeline

Developer Commit

Code is merged (master/trunk)

Stage

Stage

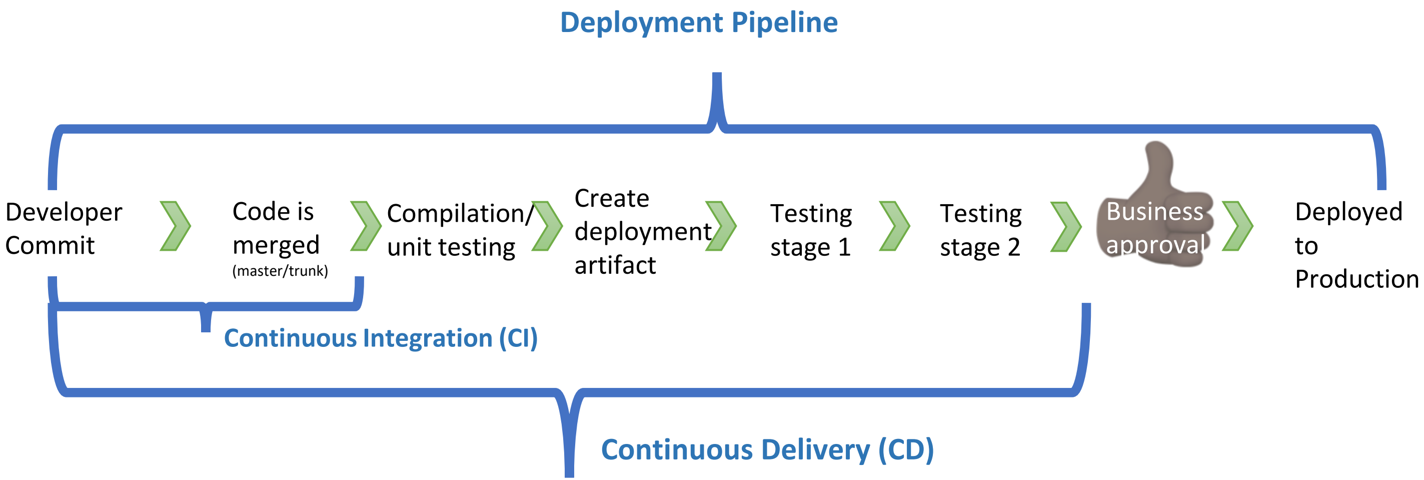
Stage

Stage

Stage

Deployed to Production

Continuous Integration (CI)



Deployment Pipeline

Developer Commit

Code is merged (master/trunk)

Compilation/ unit testing

Create deployment artifact

Testing stage 1

Testing stage 2

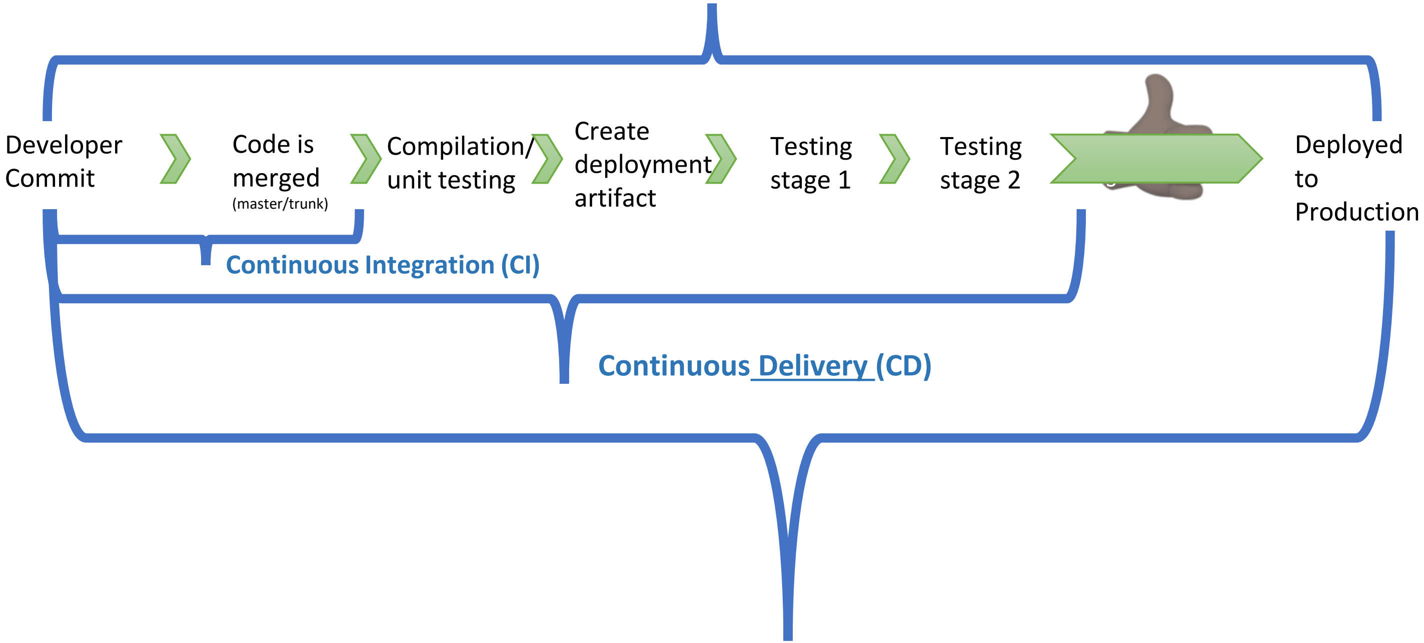
Business approval

Deployed to Production

Continuous Integration (CI)

Continuous Delivery (CD)

Deployment Pipeline



Developer Commit

Code is merged (master/trunk)

Compilation/ unit testing

Create deployment artifact

Testing stage 1

Testing stage 2

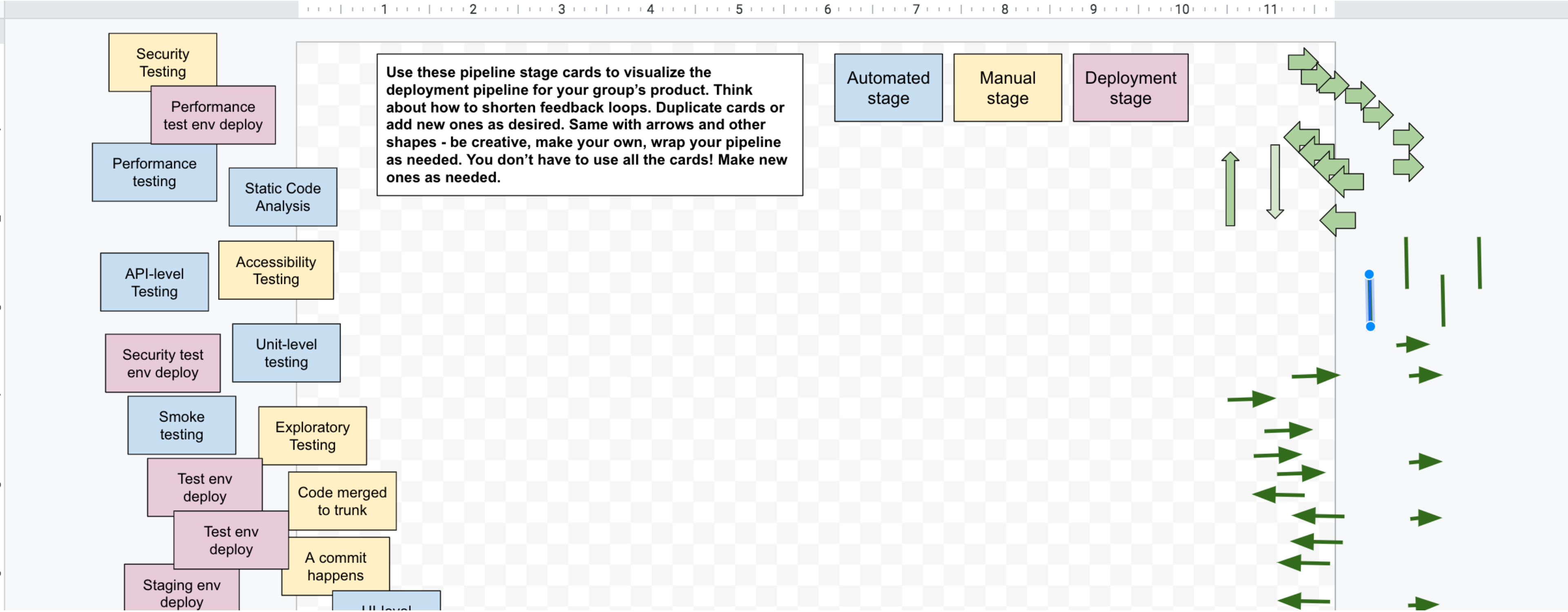
Deployed to Production

Continuous Integration (CI)

Continuous Delivery (CD)

Continuous Deployment (confusingly, this is also CD)

Demo: Visualizing your current pipeline



Use these pipeline stage cards to visualize the deployment pipeline for your group's product. Think about how to shorten feedback loops. Duplicate cards or add new ones as desired. Same with arrows and other shapes - be creative, make your own, wrap your pipeline as needed. You don't have to use all the cards! Make new ones as needed.

Try it later with your team

Make your own copy of the “Visualize Pipeline Template” from https://tinyurl.com/XA202CD into a folder shared with your team

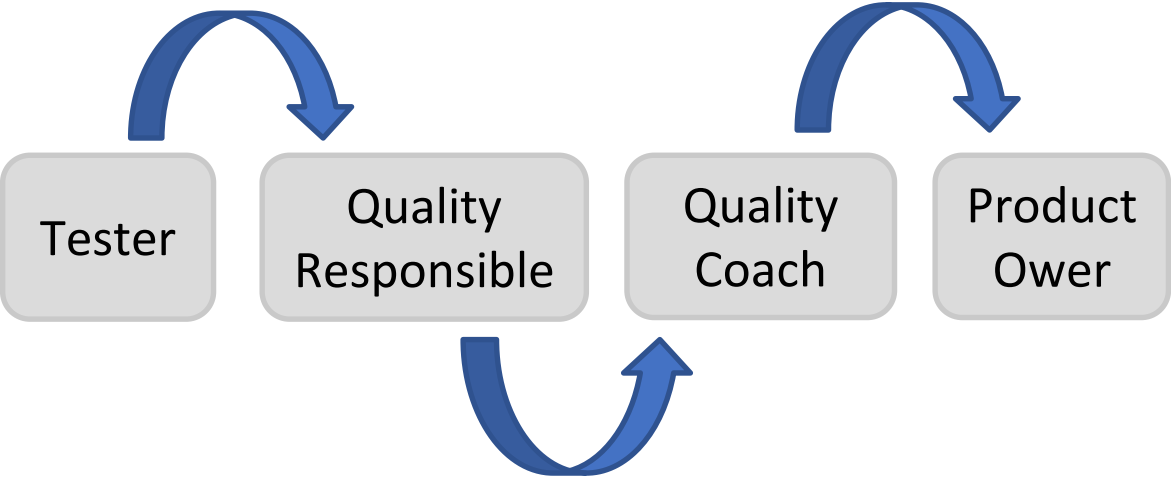
- Together in a video meeting, use the stage cards to create a model of your pipeline
- Think about how to shorten feedback loops
- You don’t have to use all the virtual step cards
- Create new step cards as needed
- Discuss!

So far we've talked about:

- The whole team approach to testing in CD
- Some terminology to help communication
- Ideas for visualizing your pipelines
- Start conversations to shorten feedback loops

Now, Areti will share more ideas to get your team engaged & improving!

About Areti



Tester

Quality Responsible

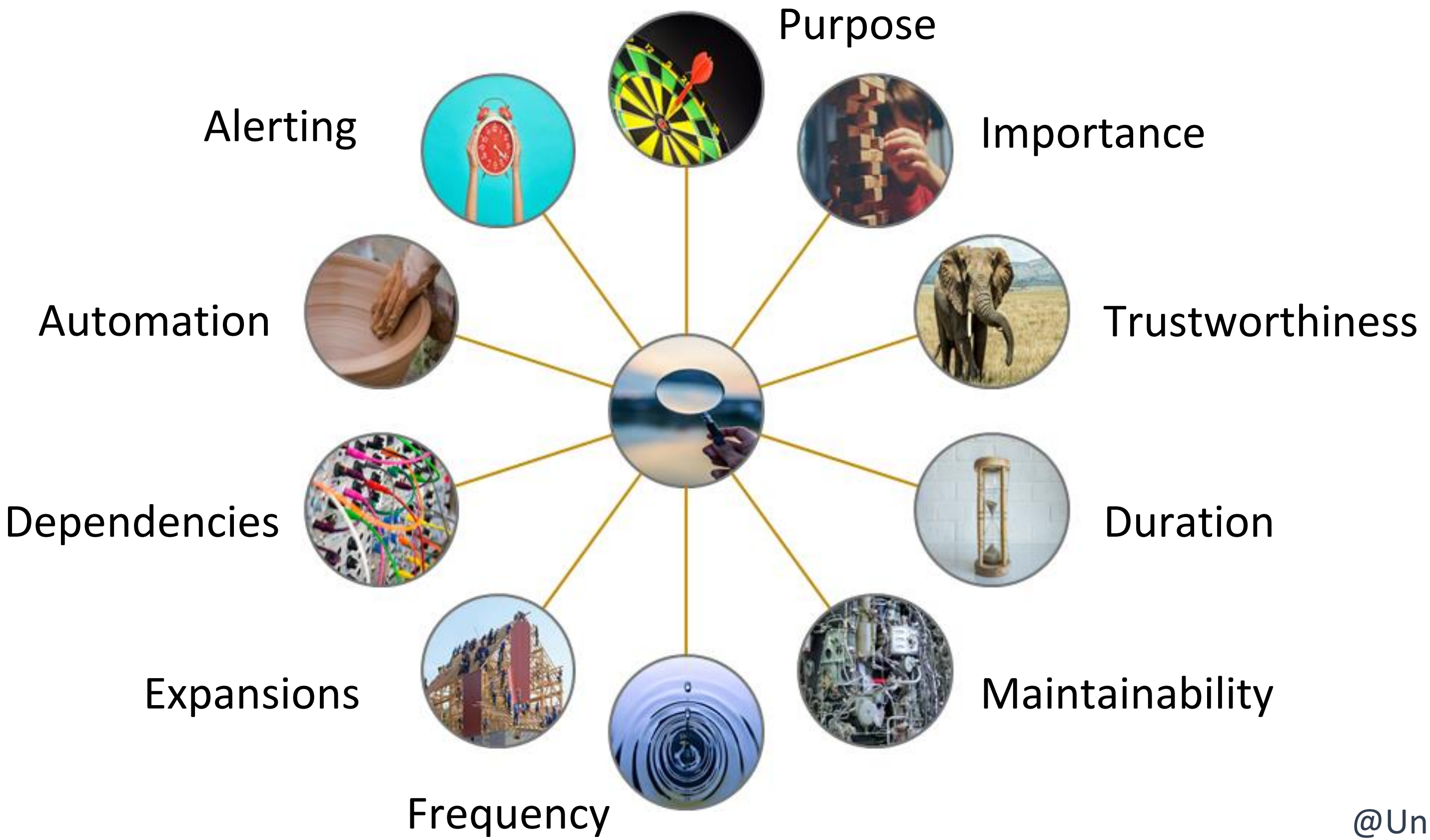
Quality Coach

Product Ower

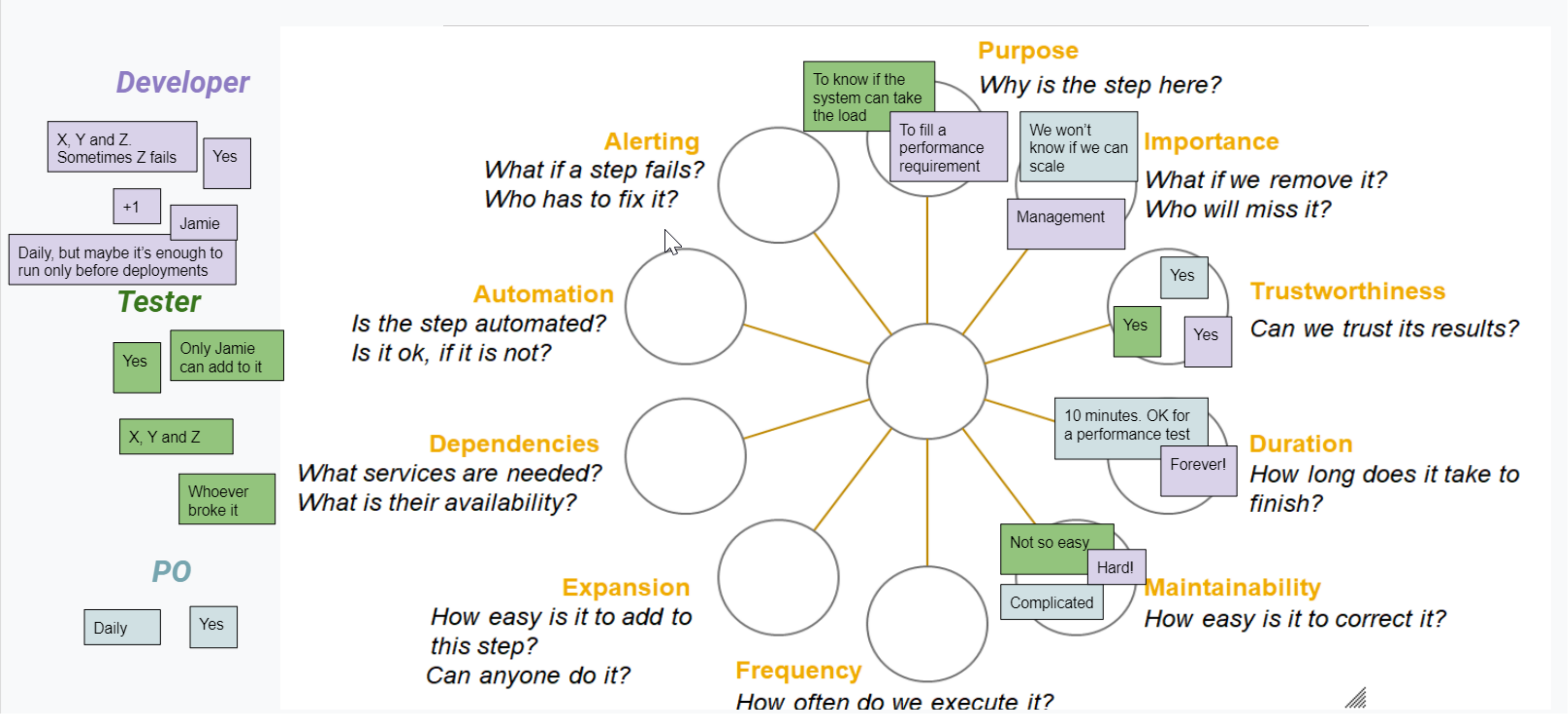
Look into your pipeline steps

- Value
- Team acceptance
- Identify constraints

Analyze each pipeline step



Demo: Analyze your pipeline steps



Developer

X, Y and Z. Sometimes Z fails

Yes

+1

Jamie

Daily, but maybe it's enough to run only before deployments

Tester

Yes

Only Jamie can add to it

X, Y and Z

Whoever broke it

PO

Daily

Yes

Purpose
Why is the step here?

To know if the system can take the load

To fill a performance requirement

Importance
What if we remove it?
Who will miss it?

We won't know if we can scale

Management

Trustworthiness
Can we trust its results?

Yes

Yes

Yes

Duration
How long does it take to finish?

10 minutes. OK for a performance test

Forever!

Maintainability
How easy is it to correct it?

Not so easy

Hard!

Complicated

Frequency
How often do we execute it?

Expansion
How easy is it to add to this step?
Can anyone do it?

Dependencies
What services are needed?
What is their availability?

Automation
Is the step automated?
Is it ok, if it is not?

Alerting
What if a step fails?
Who has to fix it?

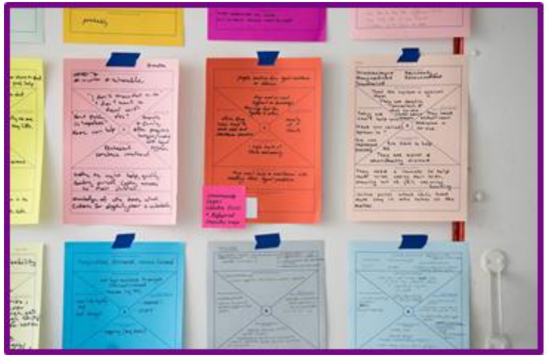
@lisacrispin

@UnremarkableQA

Try it later with your team

- Make a copy of the Analyze a Pipeline Step Exercise in https://tinyurl.com/XA202CD into a folder shared with your team
- Choose a pipeline step you want to discuss as a team.
- Each team member notes down answers to the questions in each dimension in a virtual sticky note.
- Everyone puts their answers in the dimensions.
- Discuss the results gathered in each dimension.
- Get an overview on the points that the team agrees on and the ones that can be improved.

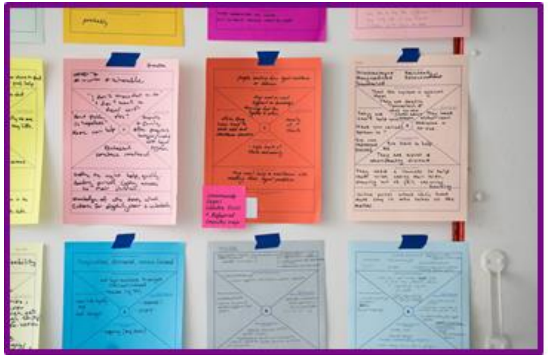
Sharing pipeline steps knowledge



<u>Identify pipeline steps</u>

Visualize all the steps in your deployment pipeline

Sharing pipeline steps knowledge



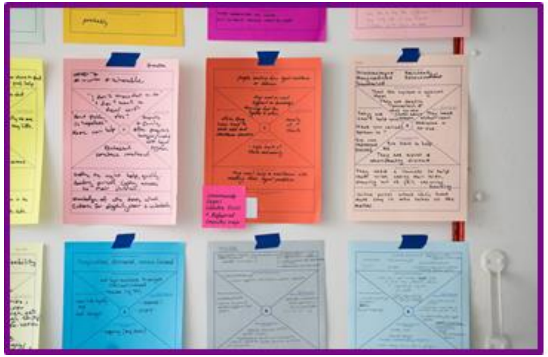
Identify pipeline steps

Visualize all the steps in your deployment pipeline

Discuss pipeline steps

Optimize the steps by engaging the knowledge of the team

Sharing pipeline steps knowledge

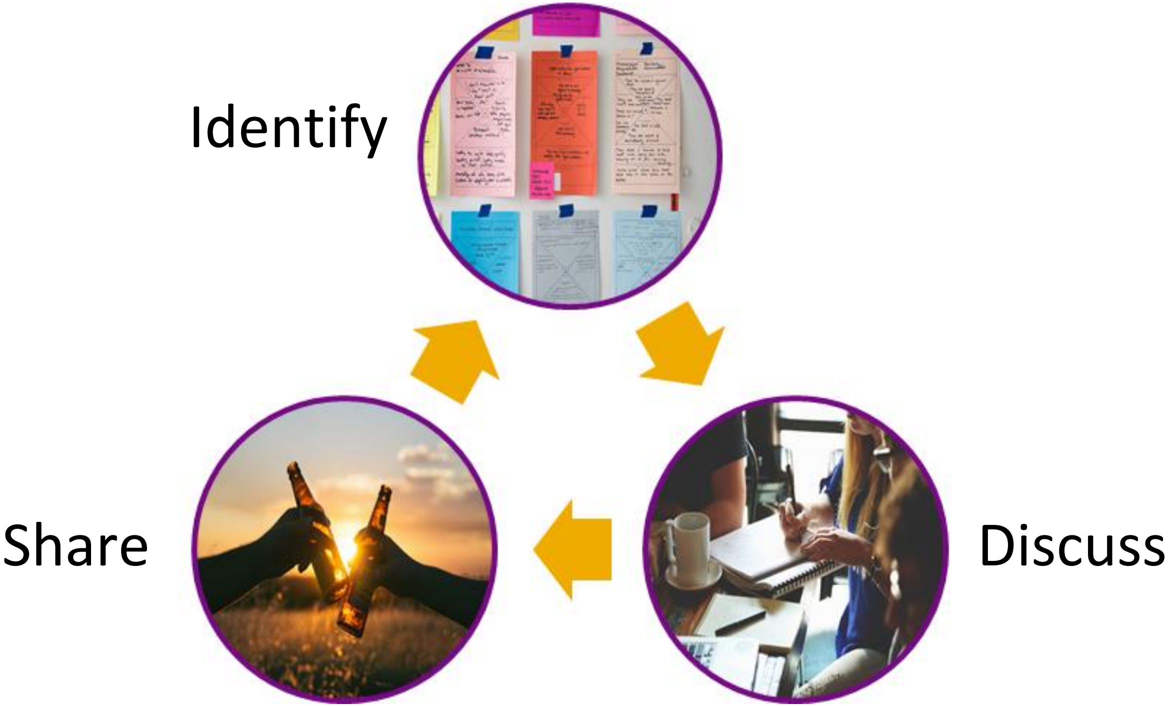


Identify pipeline steps
Visualize all the steps in your deployment pipeline

Discuss pipeline steps
Optimize the steps by engaging the knowledge of the team

Share ownership of pipeline steps
At least 2 people know each step equally well

Rinse & Repeat



Identify

Discuss

Share

"Continuous" without automation

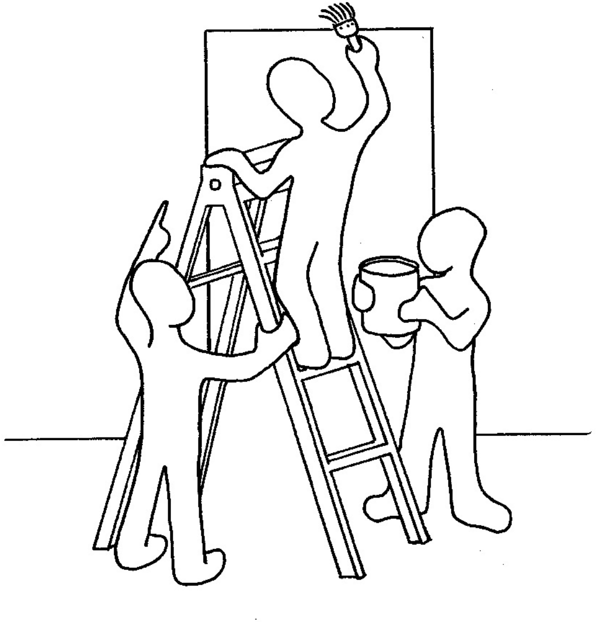
Like driving at night without your headlights, it's possible… but headlights reduce the risk.

Automated regression tests are one headlight.

Operability – monitoring, observability, testing in prod – is the other.

-- from Ashley Hunsberger

DevOps is about building a quality culture



Build relationships

- DevOps culture includes all roles
- Get to know people in and outside of your team
- Ask for help, offer help
- Do food!
- Build trust so you can learn

Show the benefits of engagement in Quality

@lisacrispin

@UnremarkableQA

Getting management support

- Speak their language – what’s the bottom line for the company?
- Help them learn why quality matters
- Help them learn the value of building quality in
- Make testing activities and their value visible

Building a quality culture

- Culture change is hard! Transforming leadership is needed
- Trust and psychological safety are prerequisites to high-performing teams
- Production ownership: the delivery team cares for the product in production
- Focus on quality, not speed, to go faster in the long run

Resources for more learning

- Testing in DevOps community site, https://testingInDevOps.org
- *Agile Testing Condensed, Agile Testing* and *More Agile Testing*, Lisa Crispin and Janet Gregory, https://agiletester.ca
- *Continuous Delivery* by Jez Humble and David Farley, https://continuousdelivery.com
- *A Practical Guide to Testing in DevOps* by Katrina Clokie
- *Accelerate* by Nicole Forsgren, Jez Humble, Gene Kim
- "Value streams are made of people", Liz Keogh, (Also see her work on BDD, Cynefin ™ & Real Options) https://lizkeogh.com/value-streams-are-made-of-people/
- *More Fearless Change* by MaryLynn Manns and Linda Rising
- Learn about behavior-driven development, example mapping: https://docs.cucumber.io/bdd/
- Trunk-based development, Paul Hammant, https://trunkbaseddevelopment.com/

Download the exercise templates & more to share with your team!

https://tinyurl.com/XA202CD

Let’s keep the conversation going...

Gratitude to Abby Bangser, Ashley Hunsberger, Elizabeth Zagroba, Janet Gregory for their contributions

Pictures credits

- Slide 28
- Slide 29a, Slide 29b, Slide 29c, Slide 29d, Slide 29e, Slide 29f, Slide 29g, Slide 29h, Slide 29i, Slide 29j
- Slide 32
- Slide 33
- Slide 34
- Slide 38
- Slide 39