Chronicles of CICD

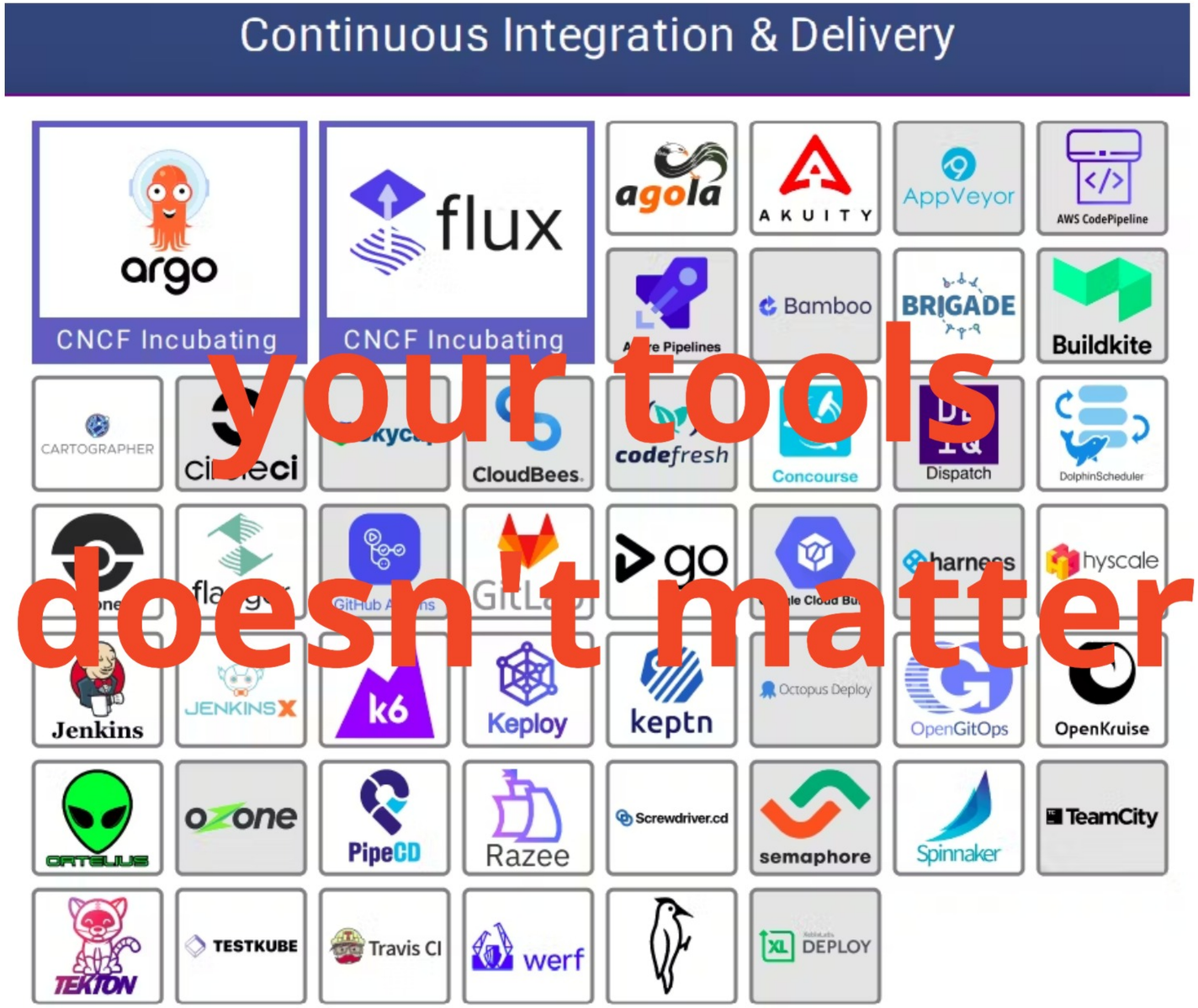
David Becvarik
heureka!group

History

How often do you integrate your code?

Why

How often do you deploy to production?


Continuous Integration & Delivery
CNCF Incubating
CNCF Incubating
your tools
doesn't matter

How

What is responsibility of your teams

Principles matters

Integrate fast

- code is always deployable
- each change is always verified
- find issues early

Automate everything

- reduce manual steps and errors
- consistency and reproducibility
- easy adoption of new tooling

Quality gating

- prevent code smell
- not able to deploy poor quality changes to production

Future

Weher are we going?

Horror stories

How did you fail?

What is right process for you?

GitFlow?

great for distributed teams, but very slow onsite. Trunk base development, extreme programming , think about it.

End2End tests

How will you develop and maintain them?
do you have good coverage on other ares?
Contract, regressions, security

Feature Flags

Brings some complexity and easy to forget.
Blue/Green deployments, Dark Launches and kill switches, did you even consider?

Everything as code

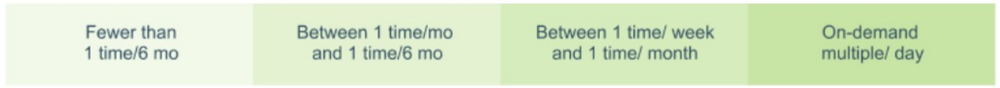
Are you level of automation practices mature enough?
Can you generate even docs?
How will you monitor drfits?

Questions?

Measure it! DORA Metrics

Deployment Frequency (DF)

How often is the code deployed to production?



Mean Time to Recovery (MT)

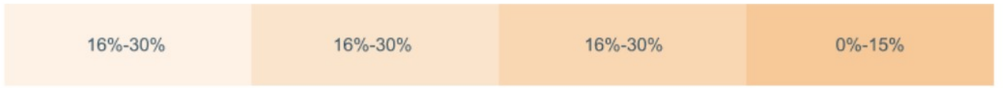
How long to restore service when there's an outage or a defect?

Lead Time for Changes (LT)

How long from code to commit to production?

Change Failure Rate (CFR)

How often deployed code degrades service?



Velocity

stability