

Increase Business Agility with Continuous Delivery

- Titles:
- Unlocking the business value of continuous delivery
- Increase business agility with continuous delivery
- Enhance business agility with continuous delivery...
- Adopt Continuous Delivery to Grow Your Business
- Continuous Delivery - the *fast path* to business agility

What is Continuous Delivery?

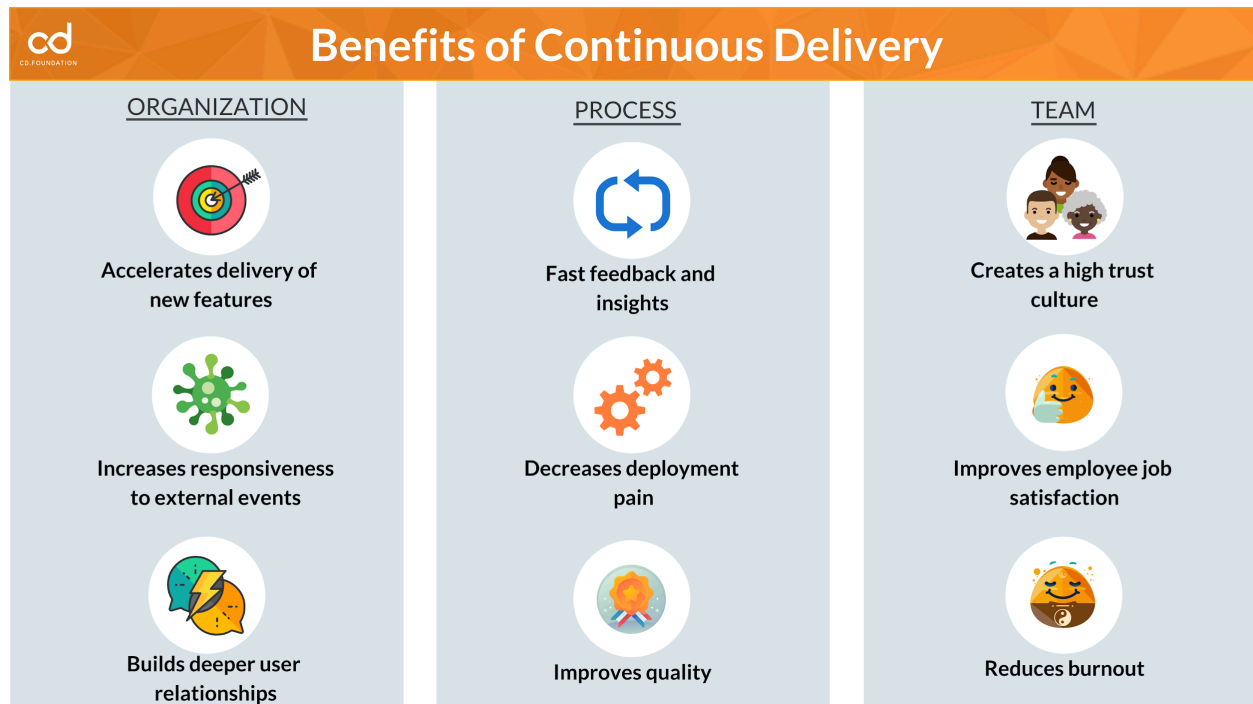
Software increasingly plays a key role in various organizations. Delivering new software features faster, while making sure that services are solid and secure is a key differentiator for every business.

Continuous Delivery is a software engineering approach in which teams produce software in short cycles, ensuring that the software can be reliably released at any time. Continuous delivery is critical to how teams deliver value. Being able to release small software changes, multiple times a day is a crucial competency for companies in the modern world.

"Continuous Delivery powers innovation at Netflix ..." – XXX, <Title>, at Netflix

Impact of Continuous Delivery

Continuous Delivery has proven benefits for your organization, its processes and your teams.



Organization Benefits

By adopting continuous delivery practices organizations benefit from

- Rapid delivery of new features
- Take advantage of fast feedback to build deeper relationships with users
- Ability to pivot when needed to respond to industry and world events
- Lack of alignment -> Aligned across business goals

"At the onset of the global pandemic, continuous delivery meant ..." – XXX,
 <Title>, at YYY

Process Benefits

Continuous delivery provides benefits for internal processes

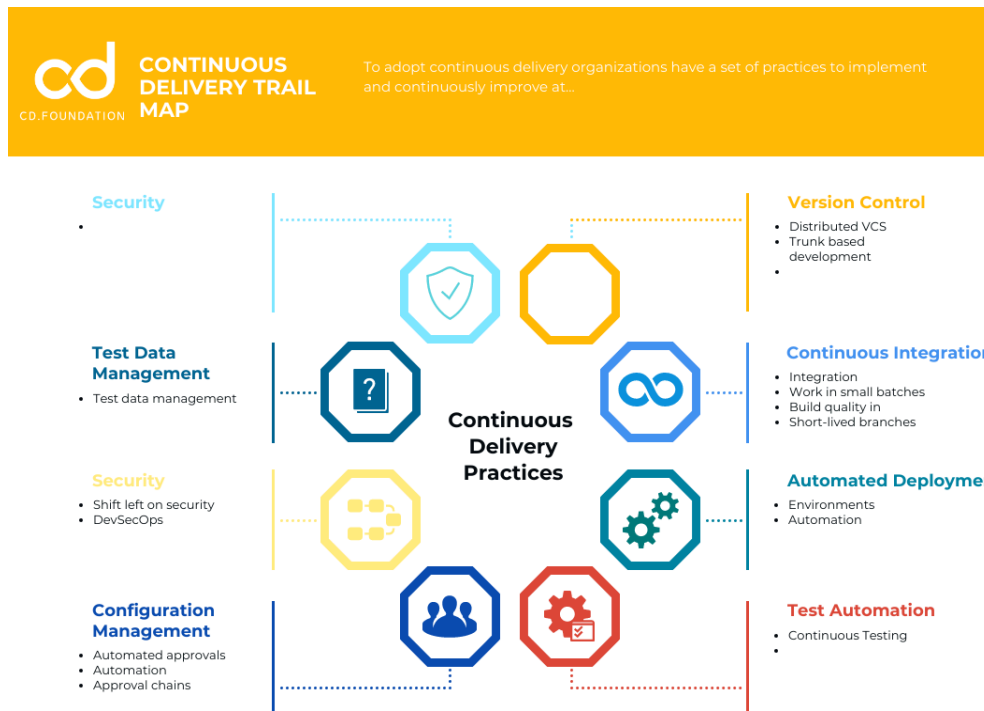
- Lots of manual work -> Mostly automated work
- Long cycle times -> Short cycle times
- Poor visibility -> Fast feedback & insight

People Benefits

- High burnout -> High job satisfaction
- Stagnant progress -> Able to grow & develop
- Checking boxes -> Creative innovation

Continuous Delivery Practices

In order to adopt continuous delivery, organizations must adopt these practices and continuously improve at them. The Continuous Delivery trail map highlights the different key practices that are needed to deliver software with security and speed.



Continuous Delivery Practices

1. Version Control (everything in CI, trunk based development, gitops (maybe??))
2. Continuous Integration (version control, small batches, merge, verify, build?)
 - a. Continuous Build
3. Automated Deployment (with verification?) (continuous deployment?)
4. Test Automation
5. Configuration Management
6. Security
7. Test Data Management
8. Policy
9. Artifact management
10. Linting / static analysis
11. Continuous Delivery (Pass it on) (environments?)

Building a test harness vs cd pipeline

Practices: elements of continuous delivery you can evaluate one at a time? Not changing year to year; constant. Supported by tooling, but not dependent on tooling.

High level: lifecycle management

KPIs (e.g. "how many can do a 10 min continuous build")

MVP for regulated industries (e.g. minimum pipeline for a startup with simple website, for cloud native, for X industry, etc.) ← referring to the list of practices

Practices = basic common denominators

Personas?

<https://devopsefficiency.com/>

Technology radar: for each practice, this is where the industry is at, here are tools you can use

CDF being able to answer: am I doing Continuous Delivery and how well am I doing at it?

Everyone is doing delivery: but how fast are they doing it? (4 DORA metrics, Accelerate breaks this down) - then what next? Best practices = help you improve the metrics (once you look at where you are at)

- What are you trying to do? How have other ppl solved these problems?

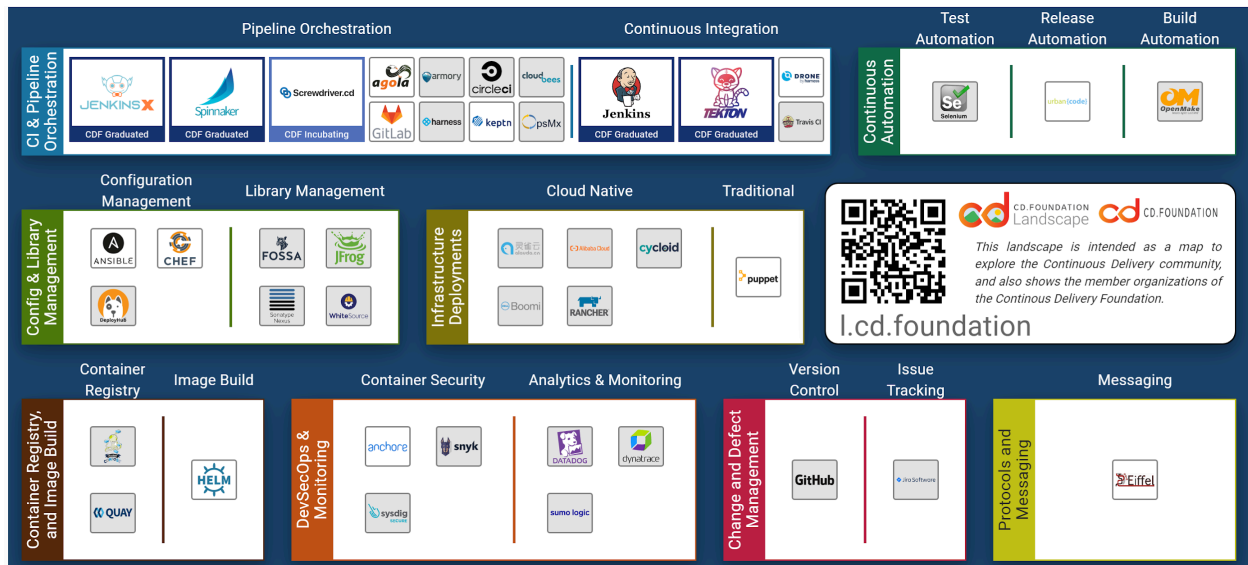
E.g. government agency with a website, services for veterans. Currently updating once a year, want to update 4 times a year. Role: tasked with getting to 4 times a year. Different conversations at VP or developer level.

Persona's:

- Christie's book: Someone wearing the build engineer hat, taking responsibility for a team's software lifecycle
- Mauricio's book: team that wants to be able to get things into customer hands faster, doesn't care about details of CI/CD pipeline
- Tracy Ragan: VP level

VP level and dev level, easier to be authoritative about the dev level (more folks to draw on)

Continuous Delivery Landscape



[CDF Landscape](#)

Continuous Delivery and Open Source

<https://github.com/todogroup/survey/blob/master/2019/chart18.svg>

Continuous Delivery in Action

1. Finance Case Study

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

2. Industry Case Study

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

3. Industry Case Study

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

- each project could share a case study - describe problem...
- tekton? ebay one

How the Continuous Delivery Foundation can help...

- Projects
- Practices
- End User Council