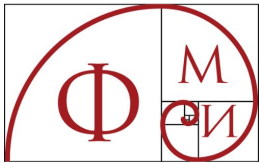
SEVERAL CLOUDS

Modern DevOps Practices

Prepared for

Faculty of Mathematics and Informatics (FMI)



Program overview

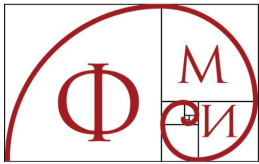
1. Initial meeting
2. **Software Development Life Cycle (SDLC)**
3. **Working with version control systems**
4. Microservices and Docker
5. Kubernetes
6. Pipelines
7. Continuous Integration
8. Continuous Delivery
9. DevSecOps
10. Cloud services in AWS
11. Infrastructure as Code with Terraform
12. Database versioning

SEVERAL CLOUDS

Software Delivery Life Cycle

Prepared for

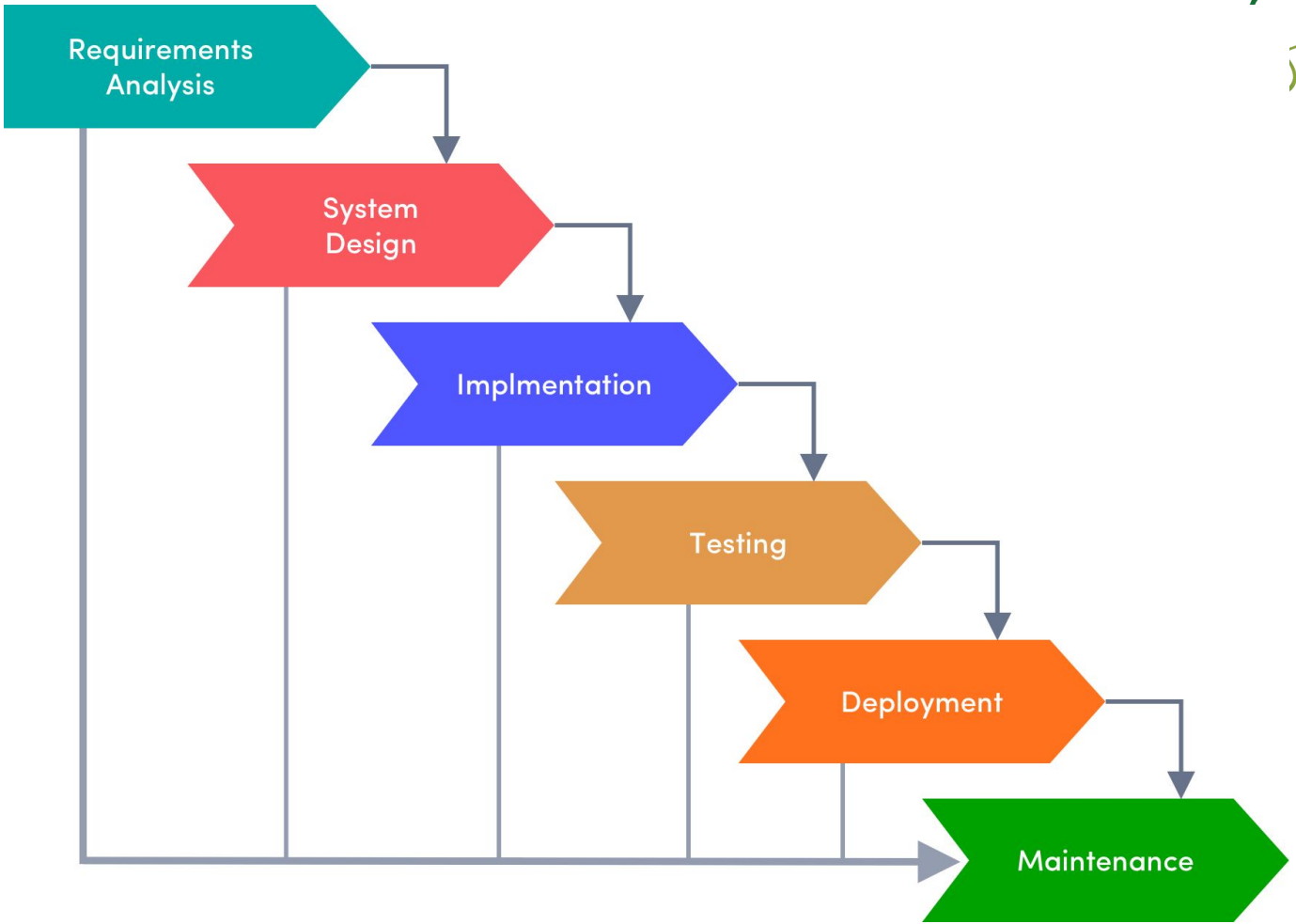
Faculty of Mathematics and Informatics (FMI)



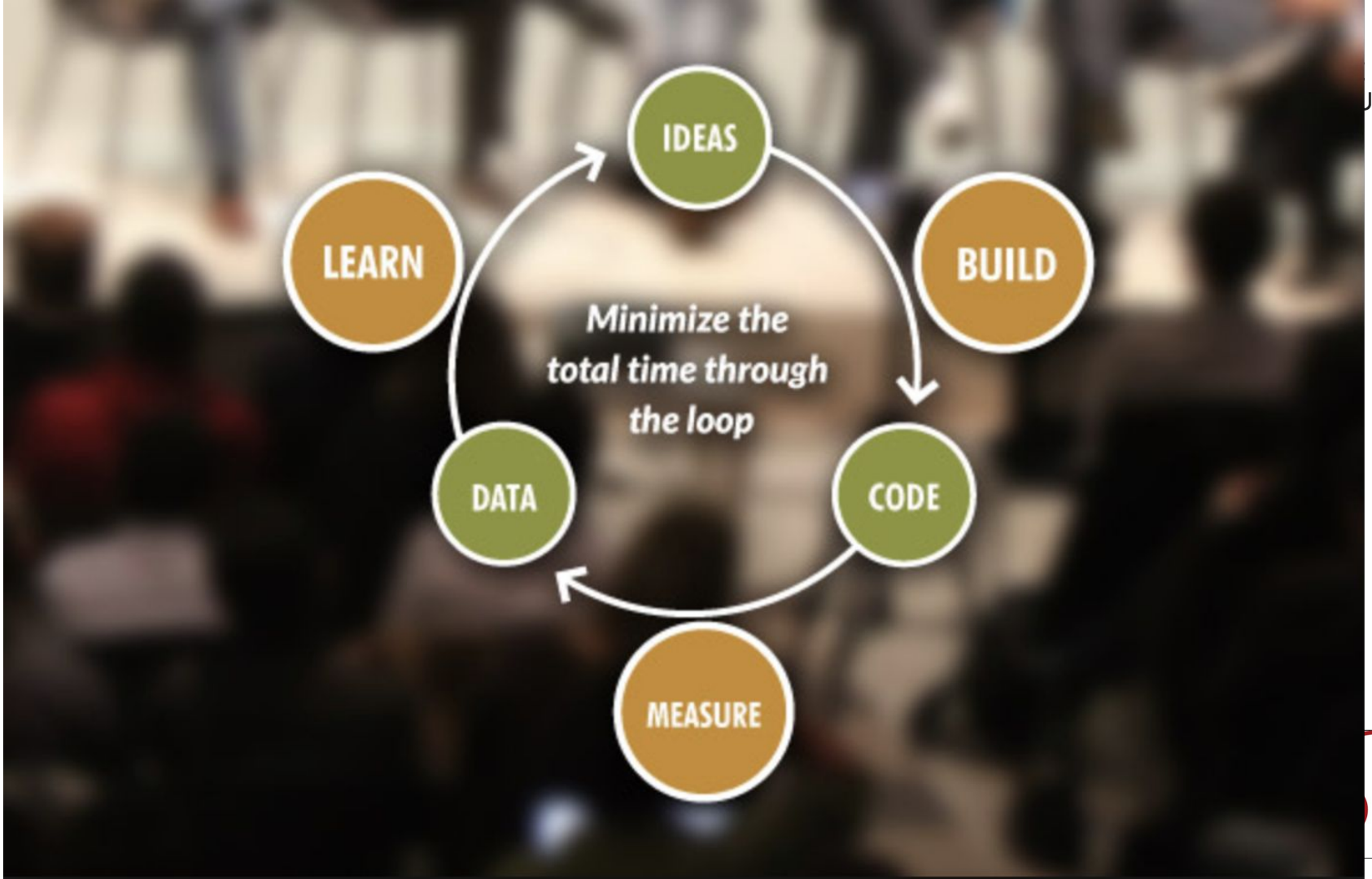
DevOps history

"You don't want to talk about the Pinto," said a Ford official. "Leave that one in the cemetery." Apparently, Ford has not forgotten the lawsuits and the public relations disasters forged by its Pinto hatchback and sedan. The Pinto's famous safety flaw, of course, was that it was prone to blowing up if rear-ended. When people talk about how bad American small cars created an opportunity for the Japanese to come in and clean house in the 1970s and '80s, they are referring to vehicles like this (and see Chevrolet Vega, second slide).

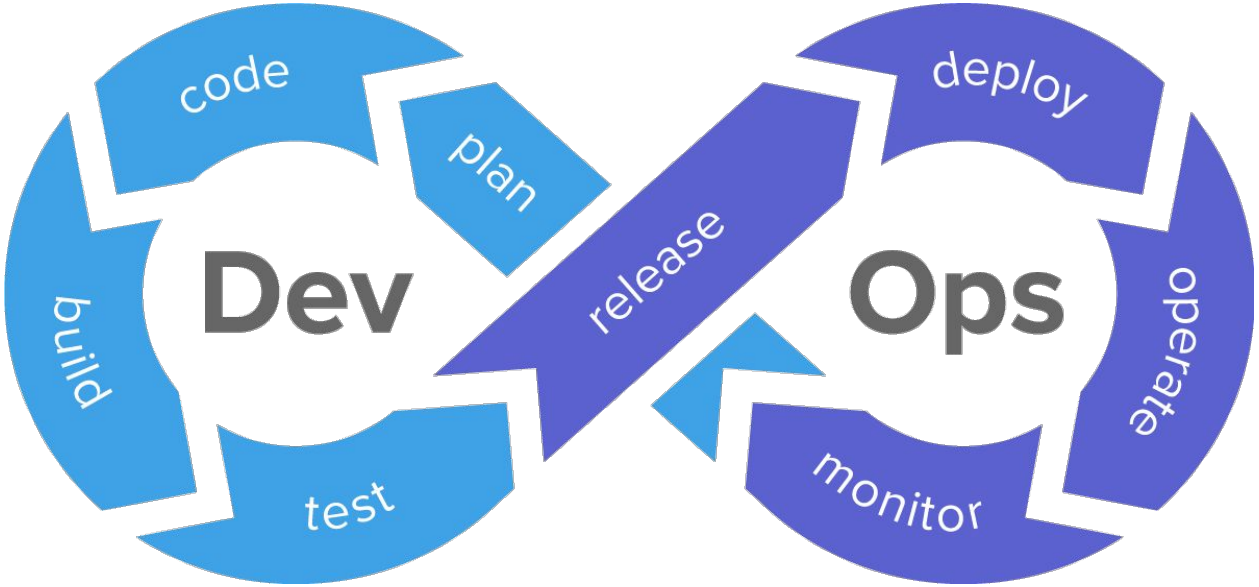
SDLC



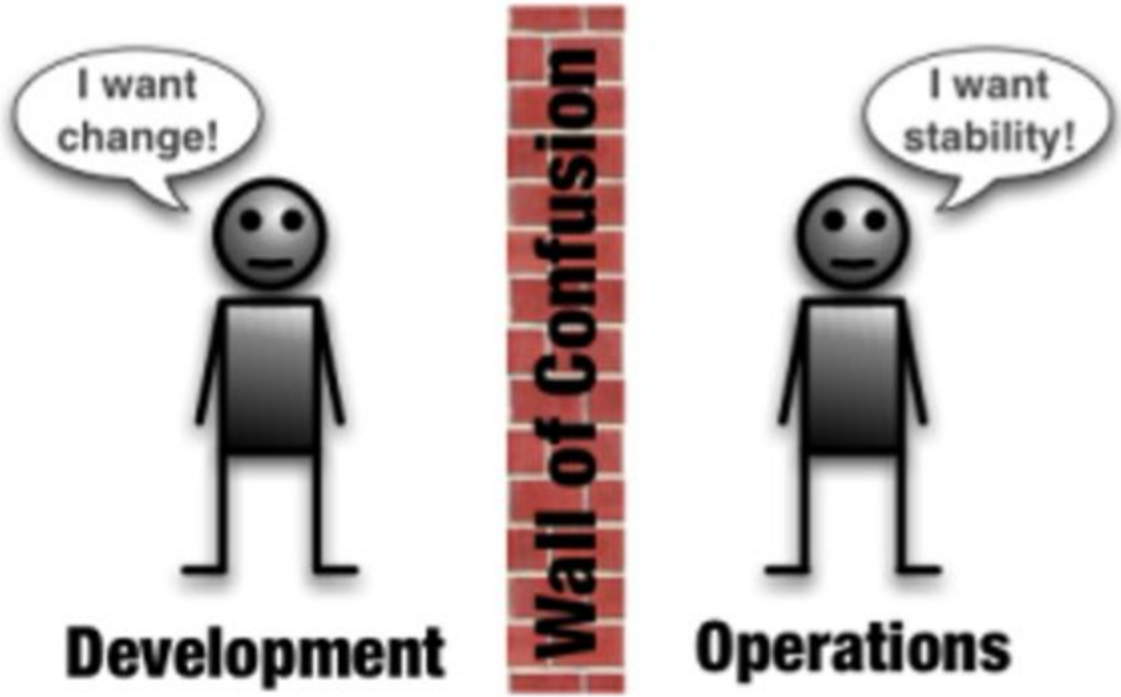
SEVERAL CLOUDS


IDEAS
BUILD
CODE
MEASURE
DATA
LEARN
Minimize the total time through the loop

SDLC


code
plan
build
test
Dev
release
deploy
operate
monitor
Ops

Wall of confusion



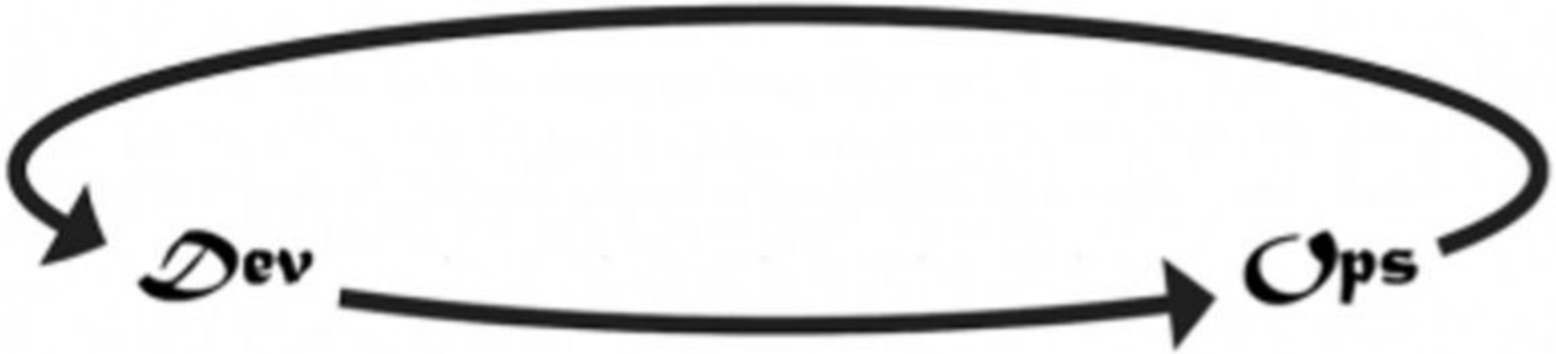
Concept to cash

SEVERAL CLOUDS

The First Way: Systems Thinking

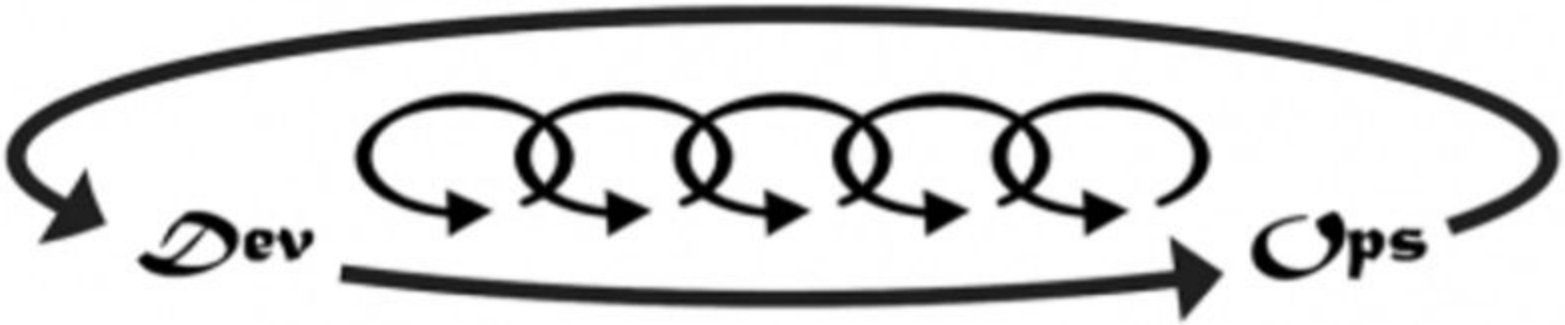
SEVERAL CLOUDS

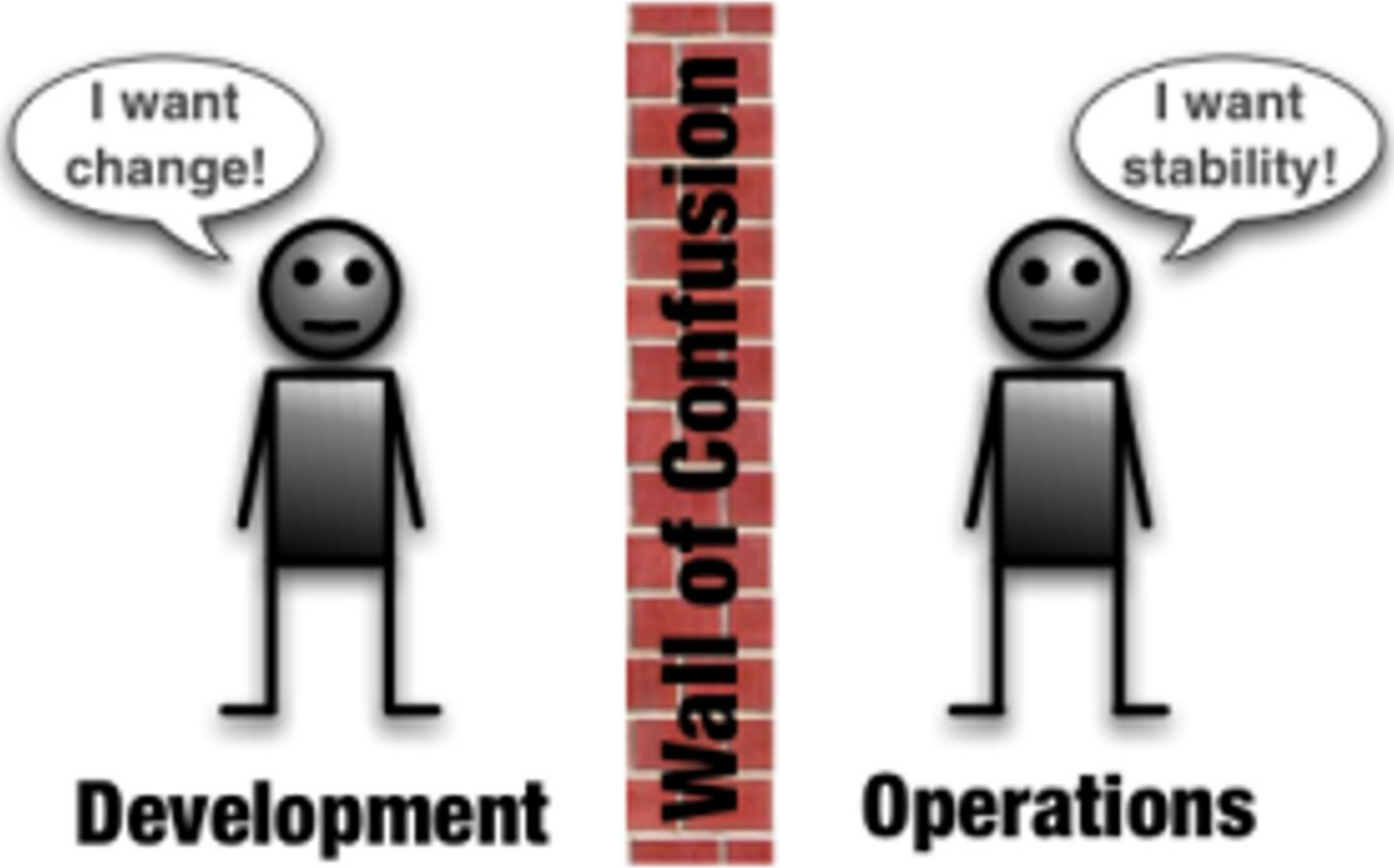
The Second Way:
Amplify Feedback Loops



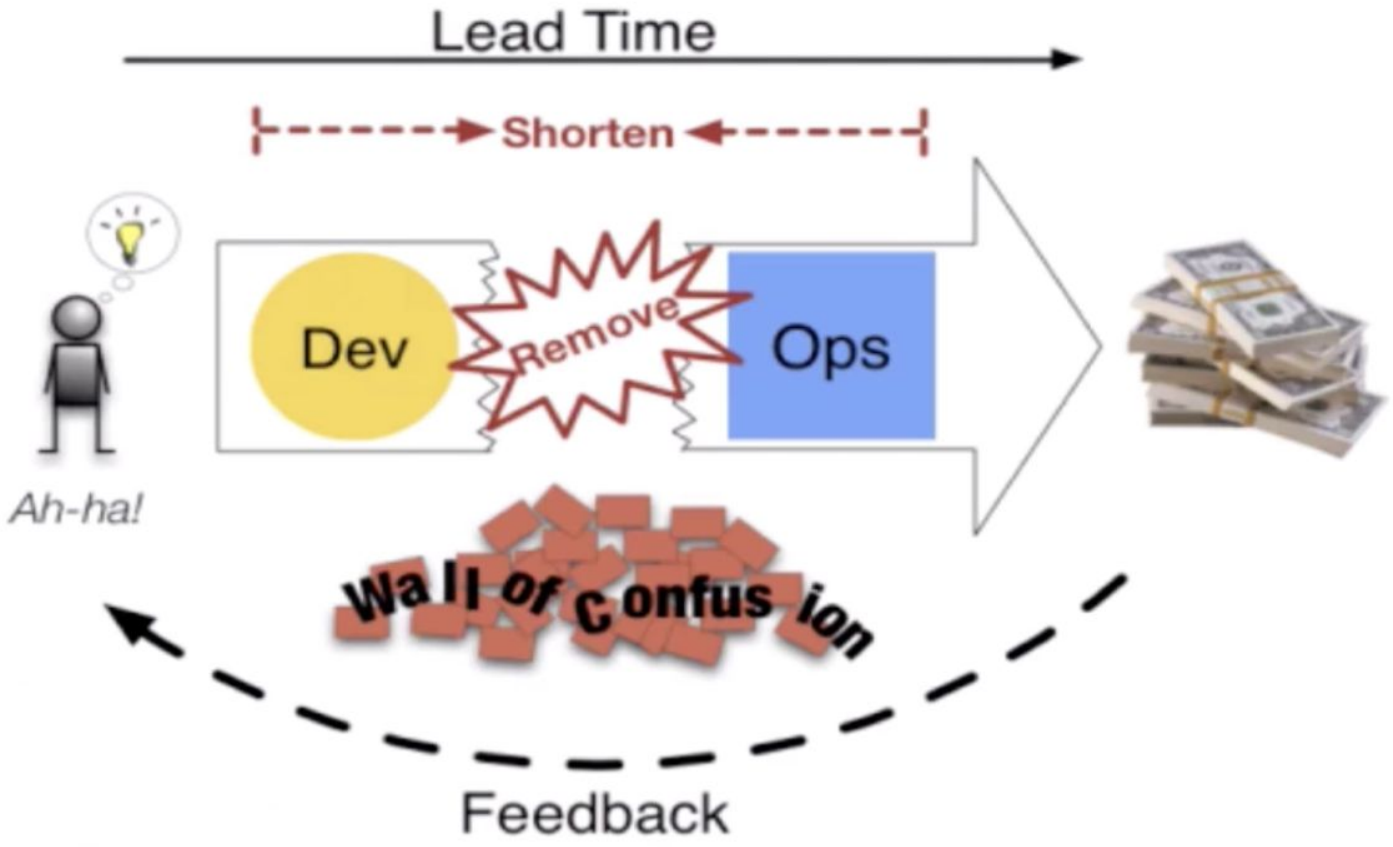
SEVERAL CLOUDS

The Third Way: Culture Of Continual Experimentation And Learning




I want change!
Wall of Confusion
I want stability!
Development
Operations

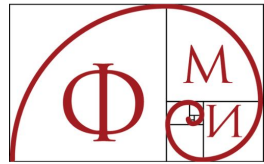
Devops Convergence



DevOps quotes

“Having a dedicated DevOps team is organizational anti-pattern.”

“Buying DevOps is like buying Agile.”



Principles behind the Agile Manifesto

We follow these principles:

Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

Business people and developers must work together daily throughout the project.

Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

Working software is the primary measure of progress.

Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

Continuous attention to technical excellence and good design enhances agility.

Simplicity--the art of maximizing the amount of work not done--is essential.

The best architectures, requirements, and designs emerge from self-organizing teams.

At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

DevOps and Agile

SEVERAL CLOUDS

ФМИ

ELITE PERFORMERS

Comparing the elite group against the low performers, we find that elite performers have…

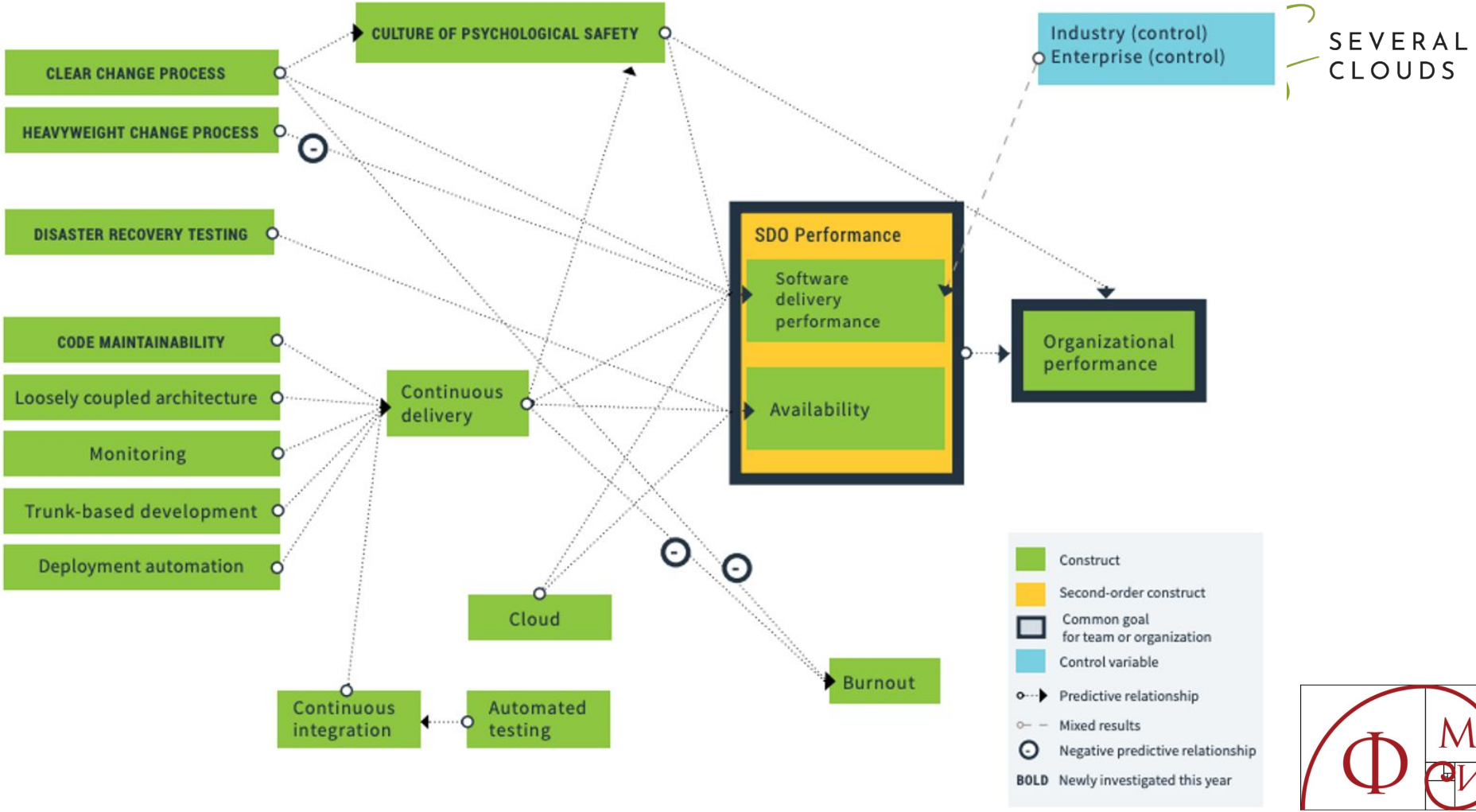
208
TIMES MORE
frequent code deployments

106
TIMES FASTER
lead time from commit to deploy

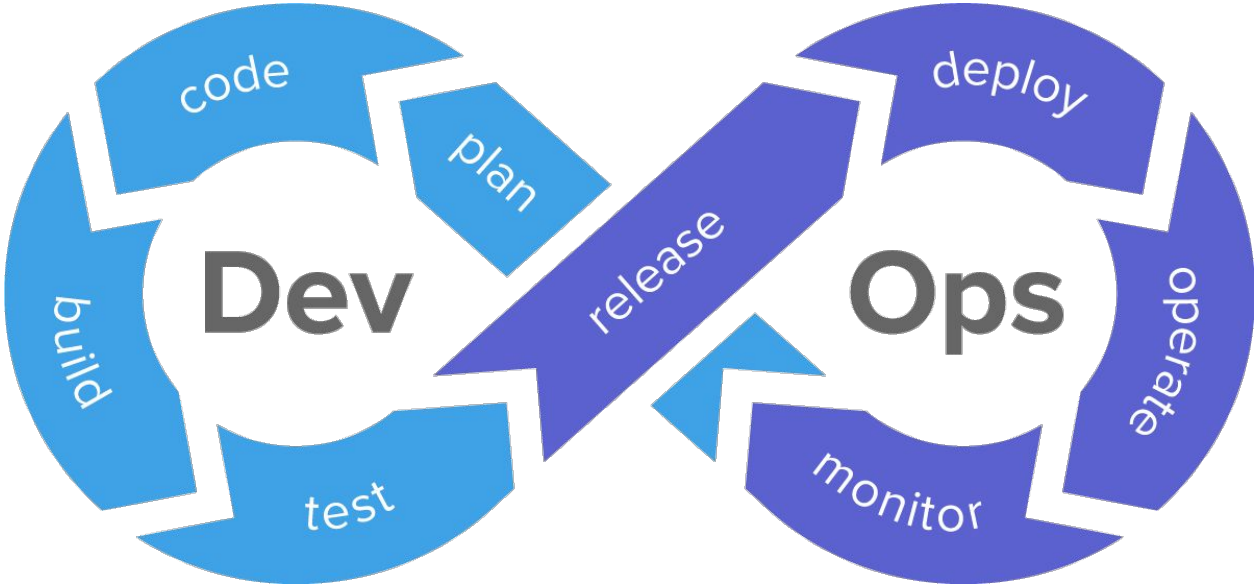
2,604
TIMES FASTER
time to recover from incidents

7
TIMES LOWER
change failure rate
(changes are 1/7 as likely to fail)

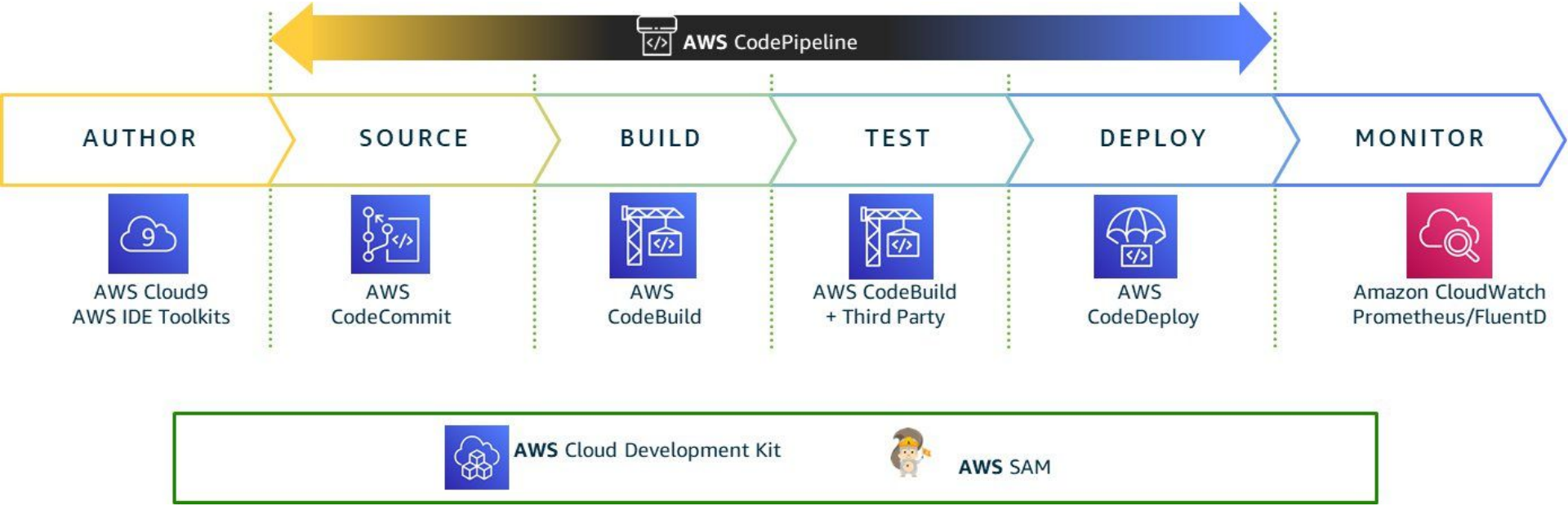
SEVERAL
CLOUDS

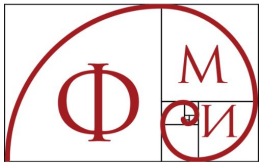

CULTURE OF PSYCHOLOGICAL SAFETY
CLEAR CHANGE PROCESS
HEAVYWEIGHT CHANGE PROCESS
DISASTER RECOVERY TESTING
CODE MAINTAINABILITY
Loosely coupled architecture
Monitoring
Trunk-based development
Deployment automation
Continuous delivery
Cloud
Continuous integration
Automated testing
Burnout
SDO Performance
Software delivery performance
Availability
Organizational performance
Industry (control)
Enterprise (control)
SEVERAL CLOUDS
Construct
Second-order construct
Common goal for team or organization
Control variable
Predictive relationship
Mixed results
Negative predictive relationship
BOLD Newly investigated this year

SDLC


code
plan
build
test
Dev
release
deploy
operate
monitor
Ops

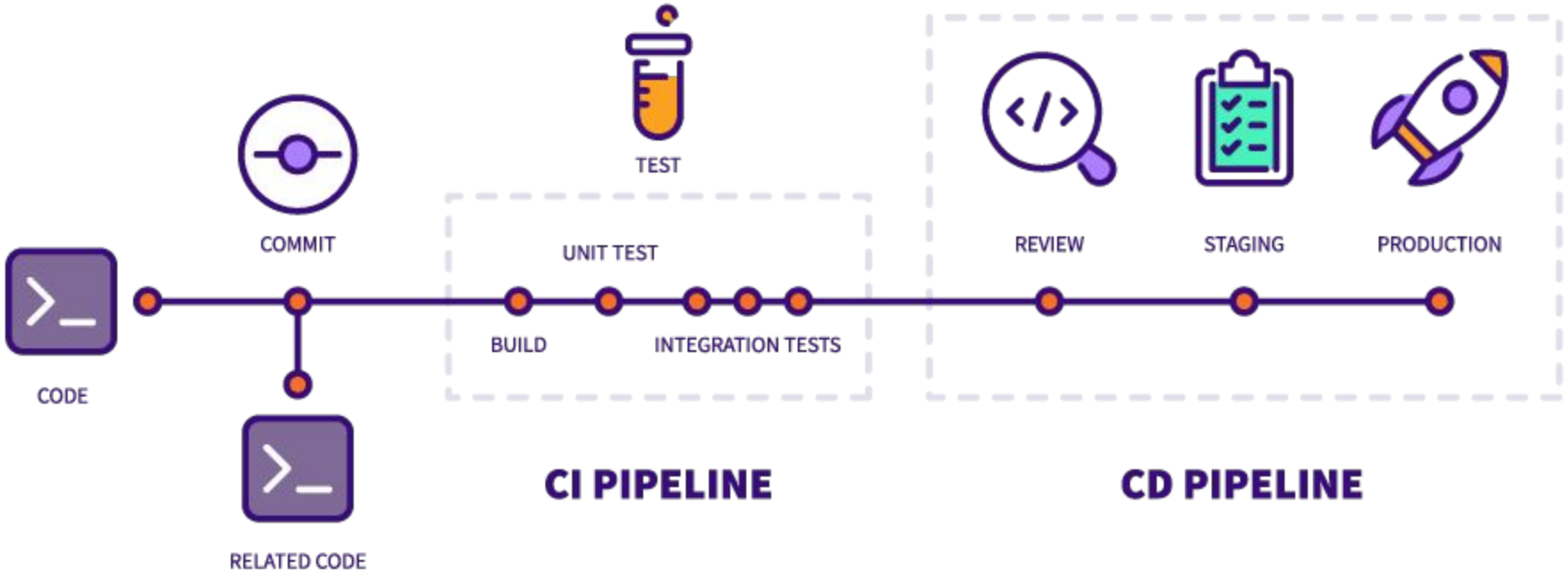
SEVERAL CLOUDS



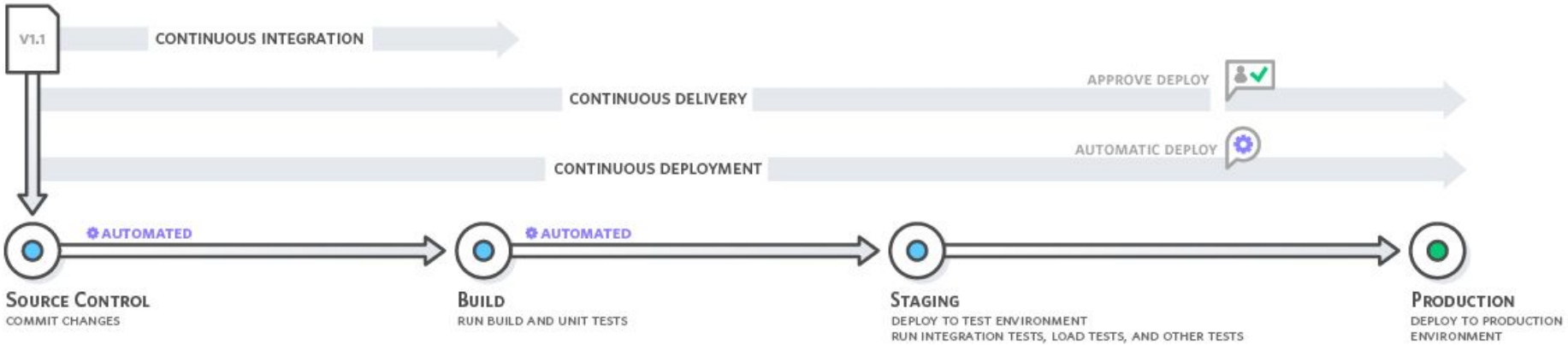


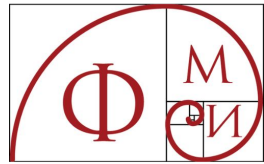
GitLab CI/CD

SEVERAL CLOUDS



SEVERAL CLOUDS



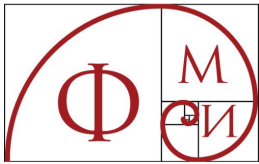


SEVERAL CLOUDS

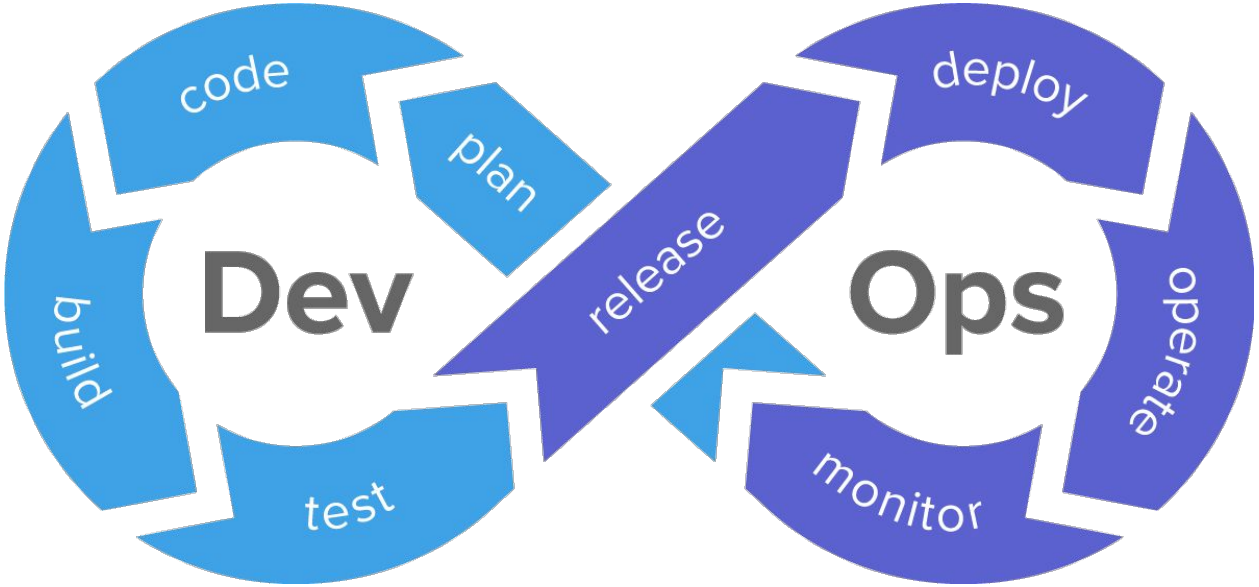
Source control management (SCM)

Prepared for

Faculty of Mathematics and Informatics (FMI)



SDLC


code
plan
build
test
Dev
release
deploy
operate
monitor
Ops

What is SCM

- tracks a running history of changes to a code base
- helps resolve conflicts
- multiple developers are working within a shared codebase

SEVERAL CLOUDS

What is Git

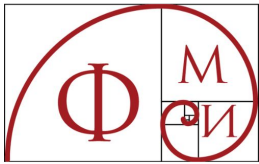
https://git-scm.com/book/en/v2/Getting-Started-What-is-Git%3F#what_is_git_section

- Everything as Code

Git Branches

SEVERAL CLOUDS

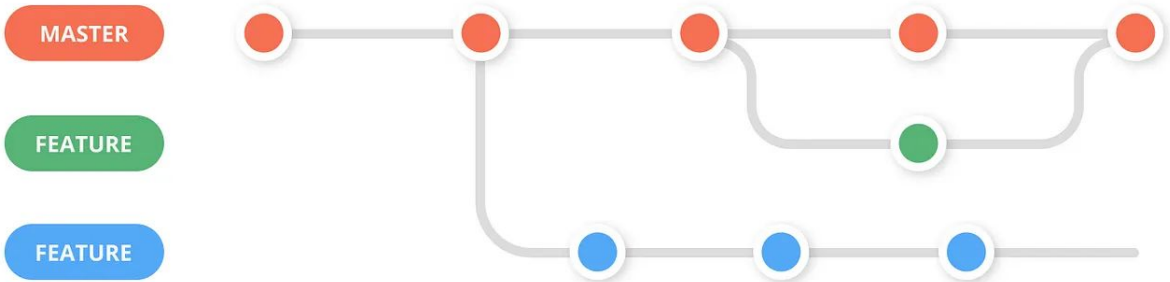
https://git-scm.com/book/en/v2/Git-Branching-Branches-in-a-Nutshell
https://svikashk.medium.com/5-git-workflows-you-can-use-to-deliver-better-code-9b8c84e0135f



Basic Git Workflow

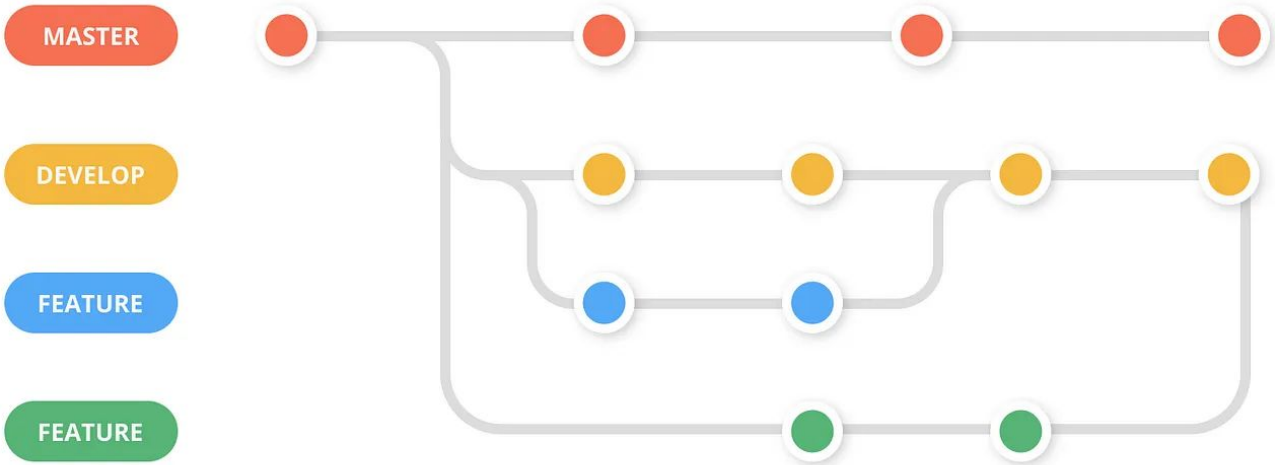
Git Feature Branch Workflow

SEVERAL CLOUDS



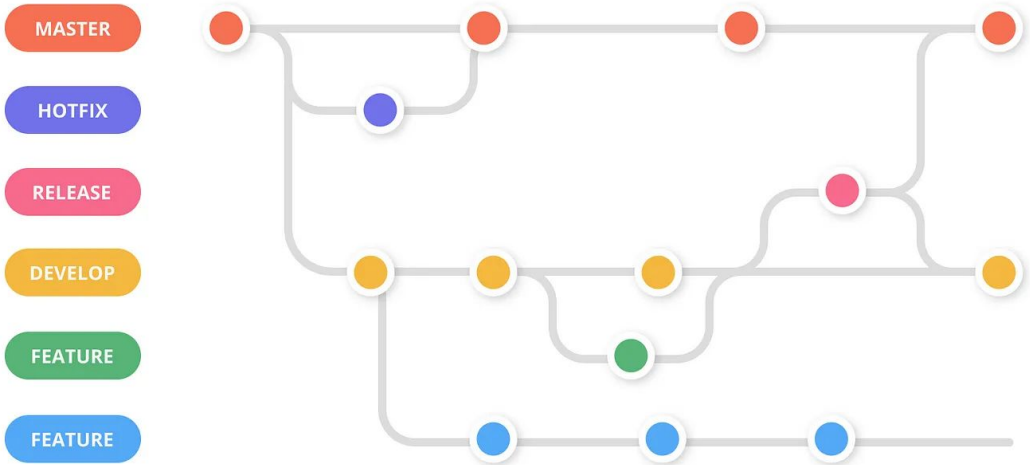
Git Feature Workflow with stable branches

SEVERAL CLOUDS

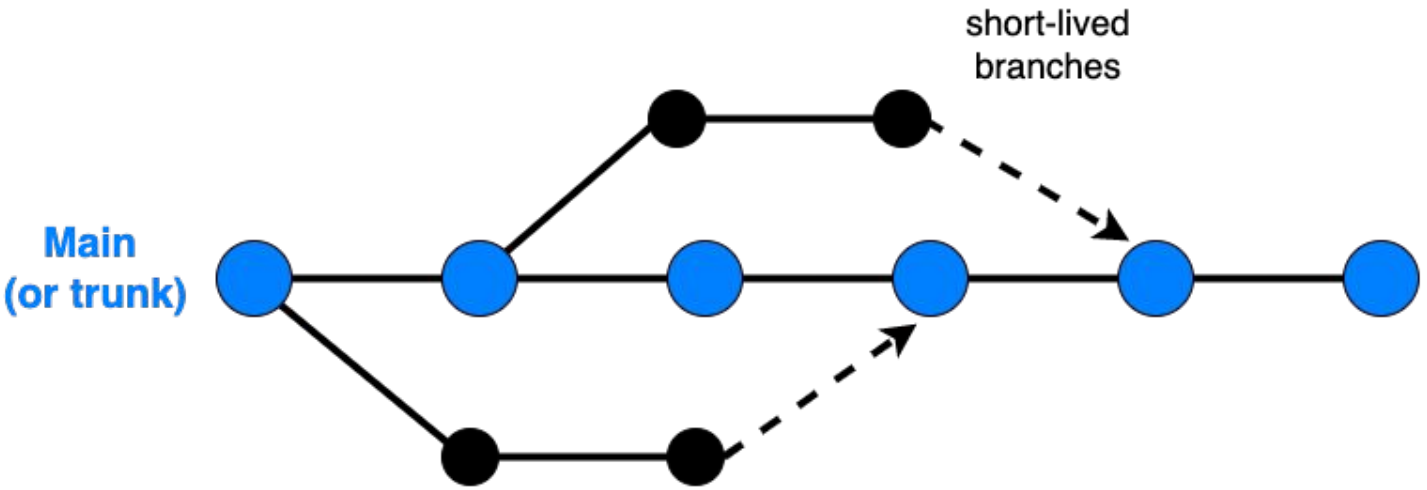


Git Flow

SEVERAL CLOUDS



Trunk-based development



merging is done more frequently and more easily
for shorter branches

Q & A

Resources

DevOps report 2023 -
https://cloud.google.com/blog/products/devops-sre/announcing-the-2023-state-of-devops-report
https://www.atlassian.com/git/tutorials/source-code-management
https://www.atlassian.com/git/tutorials/what-is-version-control
https://trunkbaseddevelopment.com/

SEVERAL CLOUDS

Thank you!

daniel@severalclouds.com
https://www.linkedin.com/in/danielrankov/