

Demystifying CI/CD

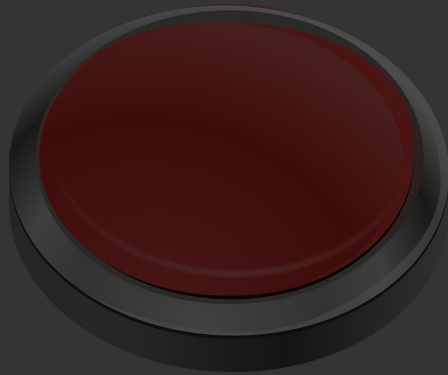
Imagine you're getting ready to ship something...



**You press the big red
button**



click





silence

Do you...

a) Sweat and wait to see if it blows up

Do you...

- a) Sweat and wait to see if it blows up
- b) Pack up, go home & enjoy your weekend

It all boils down to automation.



Image source: Forbes





Image source: Horizon Zero Dawn



PEBKAC

Tech support can't help you when you're the problem.

Image source: Demotivational

 @IAmJerdog



Director, DevRel & Community



Timeline.jerdog.me



[IAmJerdog](https://twitter.com/IAmJerdog)



[jerdog](https://dev.to/jerdog)



[/in/jeremymeiss](https://in.linkedin.com/in/jeremymeiss)

“DEVOPS”



The DevOps Lifecycle

- Plan

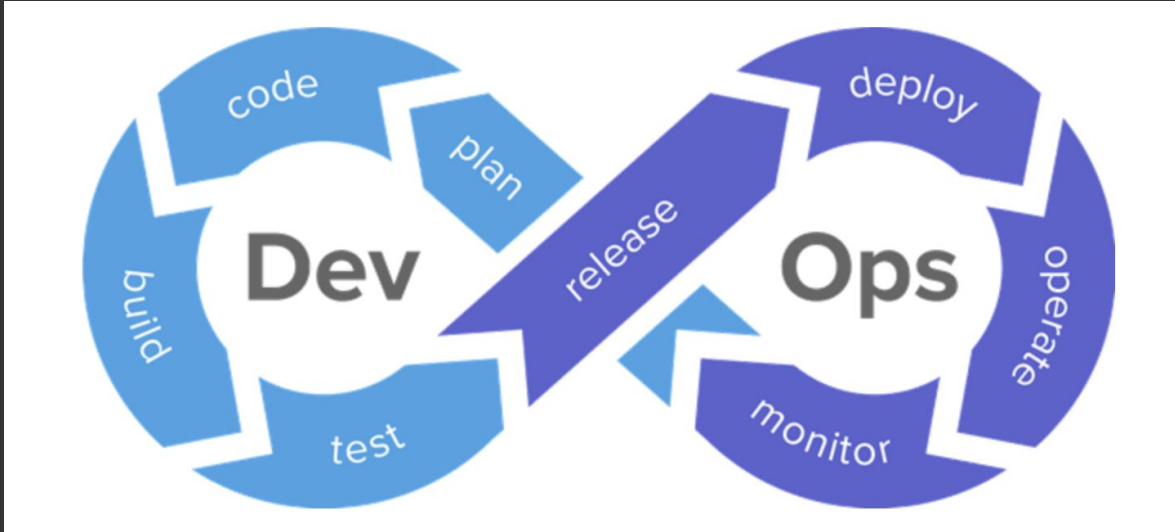


Image source: [C-Sharp Corner](#)

The DevOps Lifecycle

- Plan
- Code

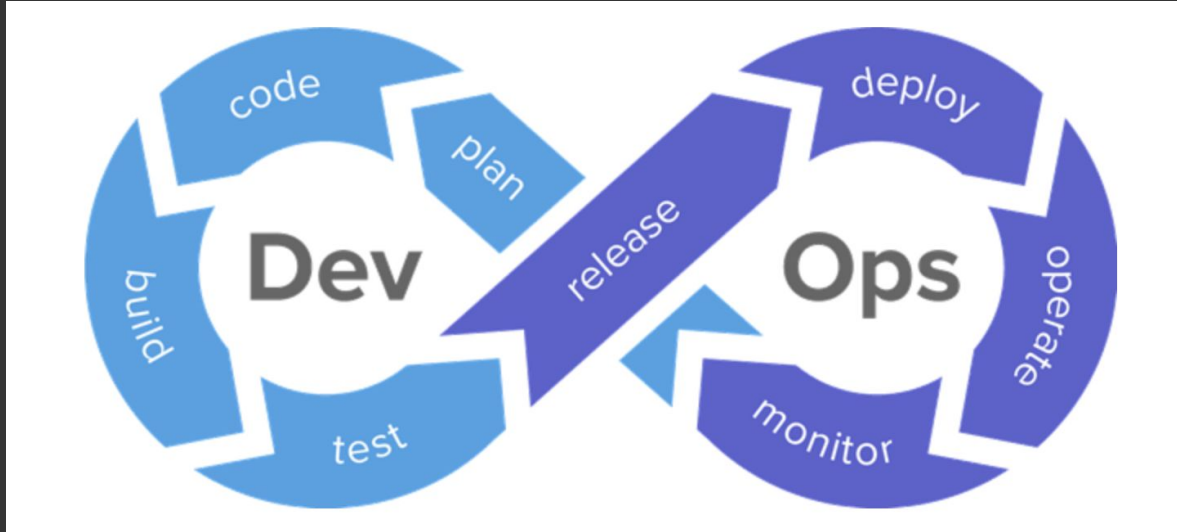
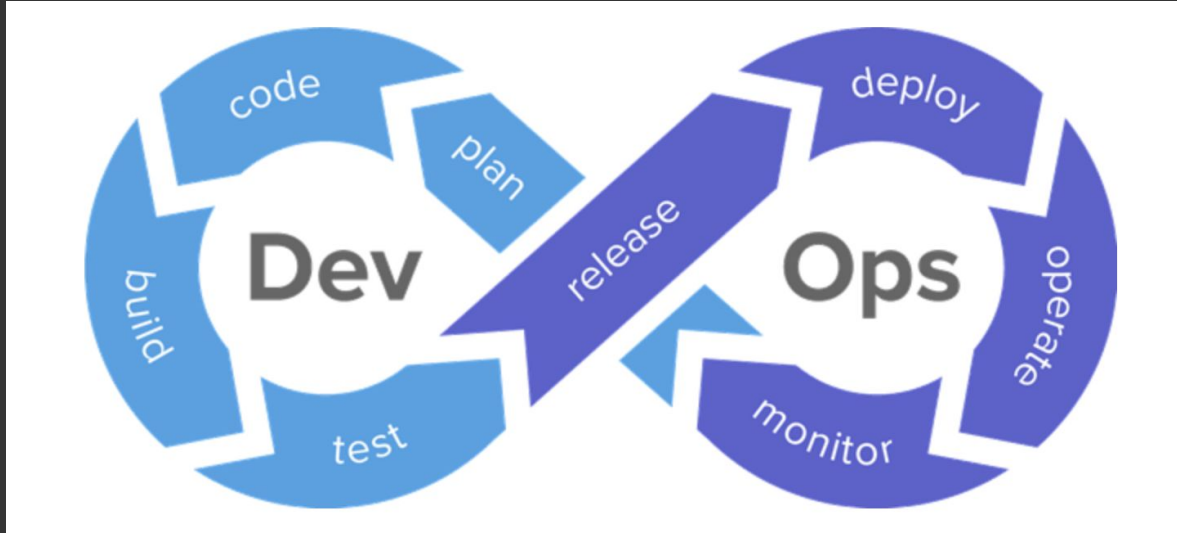


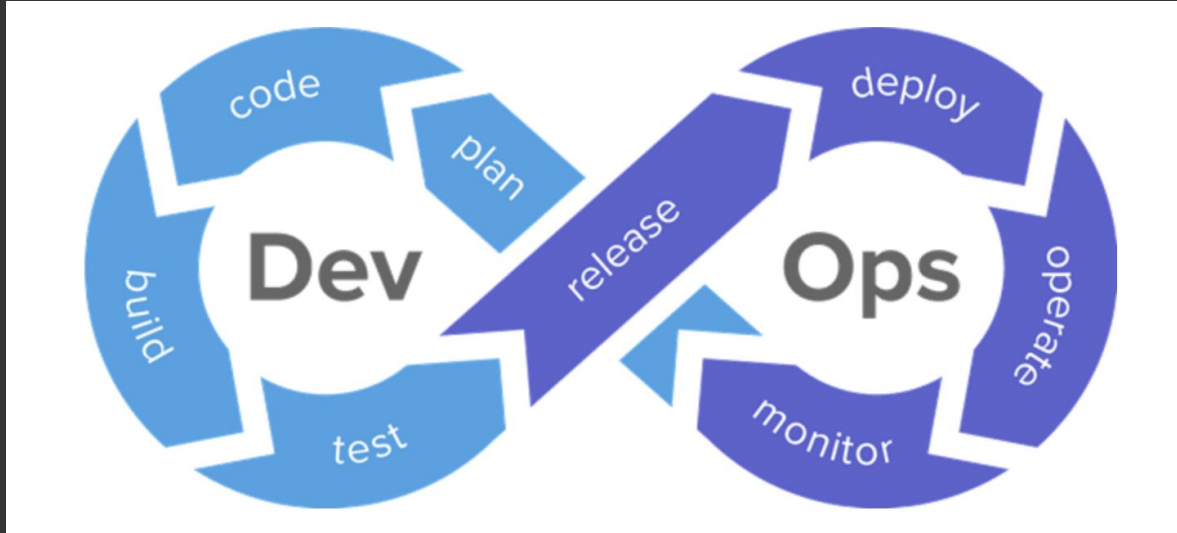
Image source: [C-Sharp Corner](#)

The DevOps Lifecycle



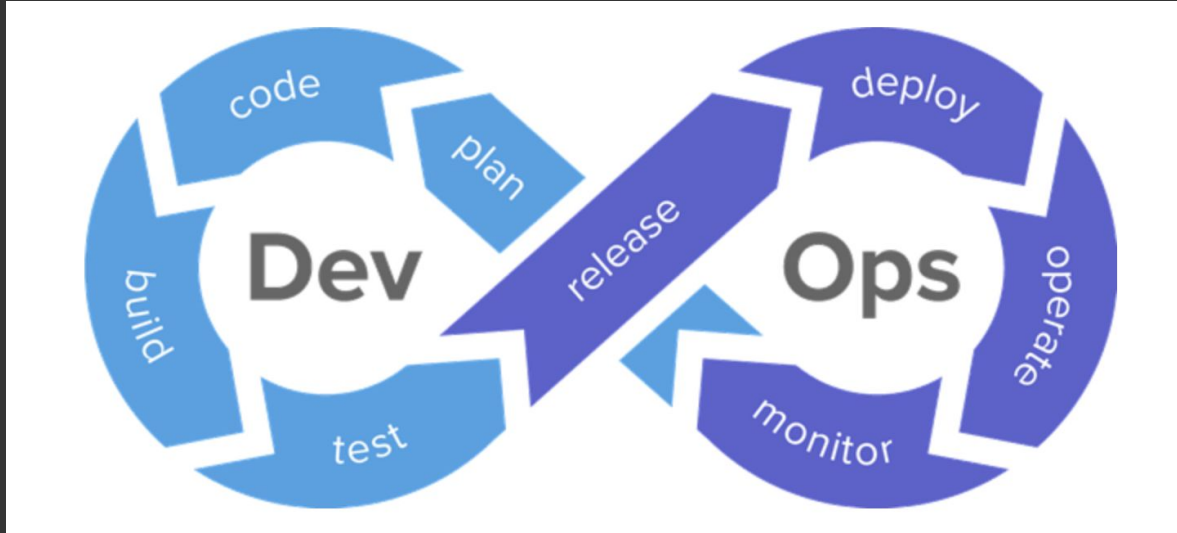
- Plan
- Code
- Build

The DevOps Lifecycle



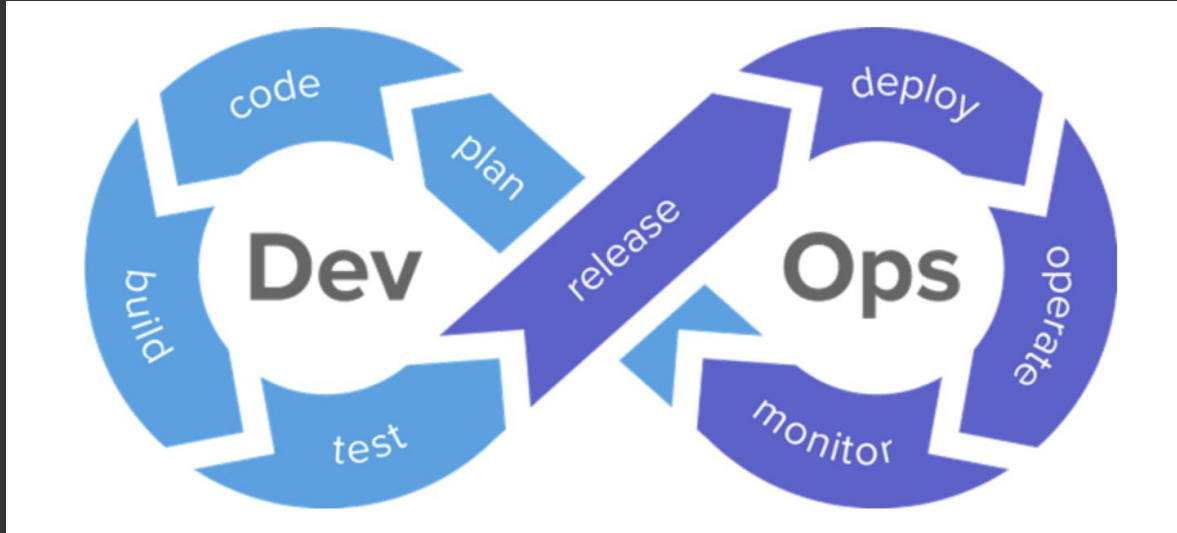
- Plan
- Code
- Build
- Test

The DevOps Lifecycle



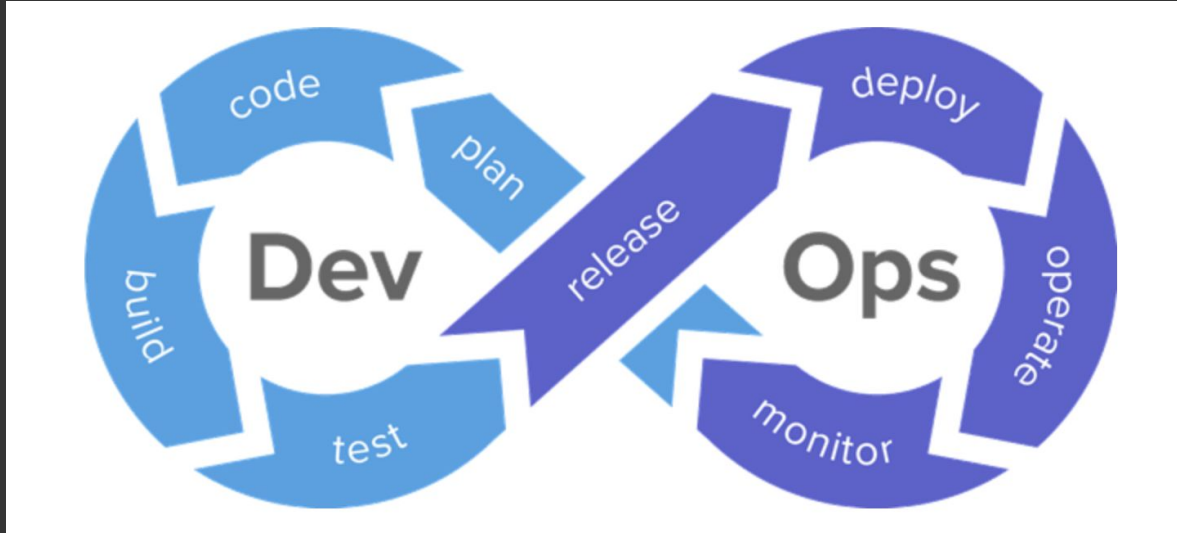
- Plan
- Code
- Build
- Test
- Release

The DevOps Lifecycle



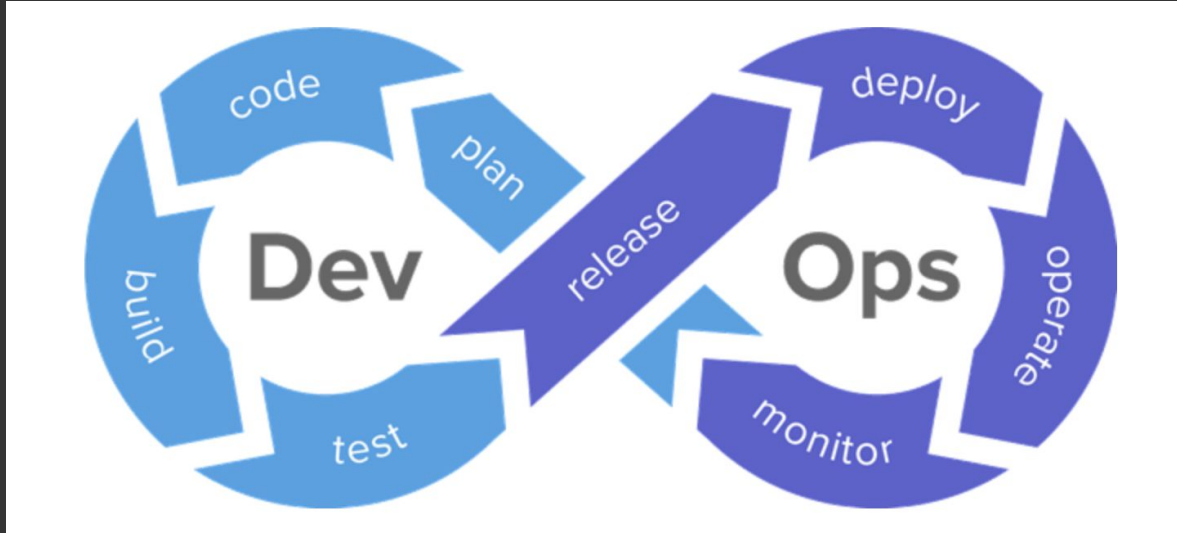
- Plan
- Code
- Build
- Test
- Release
- Deploy

The DevOps Lifecycle



- Plan
- Code
- Build
- Test
- Release
- Deploy
- Operate

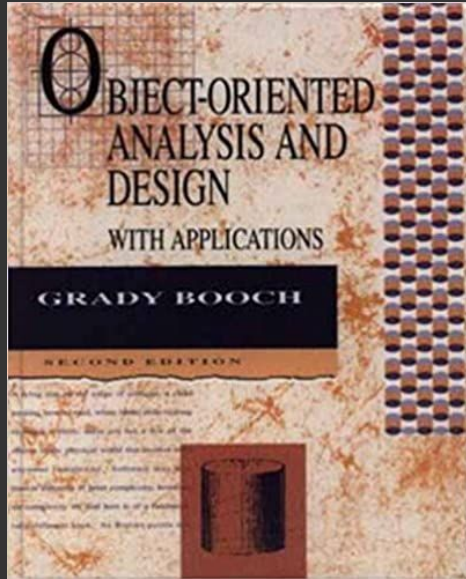
The DevOps Lifecycle



- Plan
- Code
- Build
- Test
- Release
- Deploy
- Operate
- Monitor

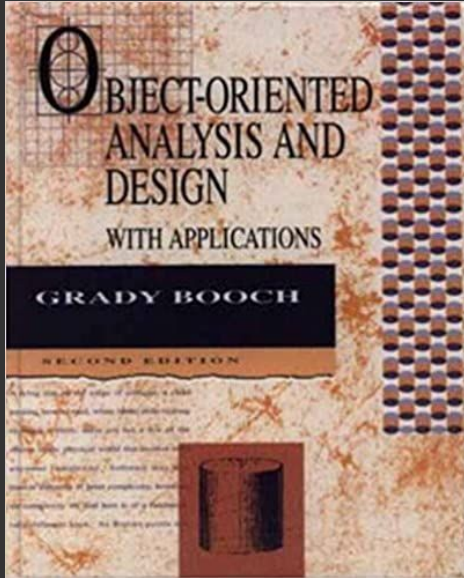
Continuous {Integration|Deployment}

Quick History of CI/CD

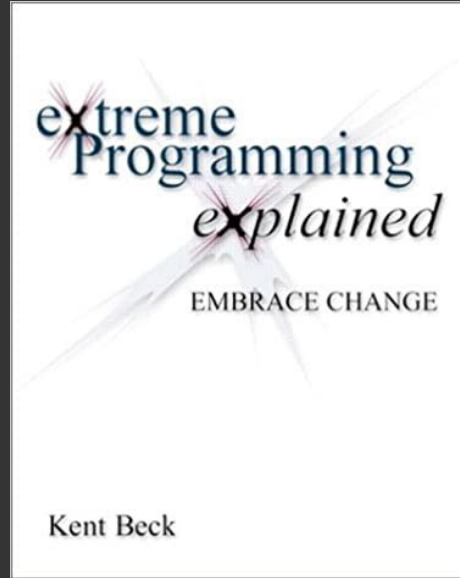


1991

Quick History of CI/CD

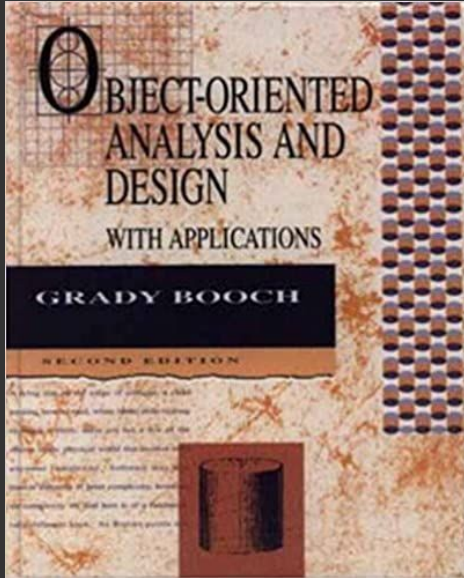


1991

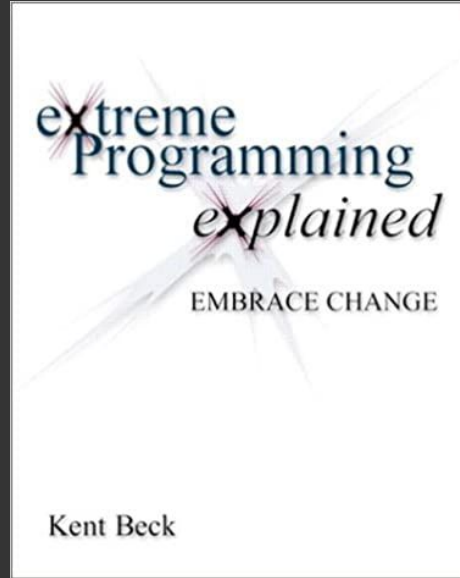


1997

Quick History of CI/CD



1991



1997



2001

Quick History of CI/CD

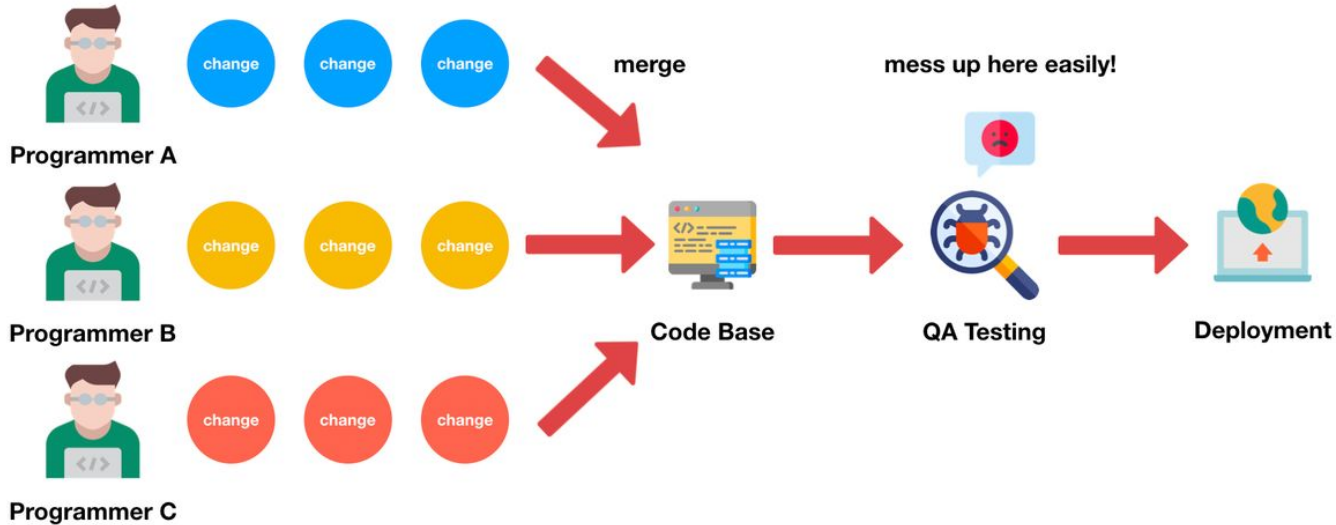


Quick History of CI/CD



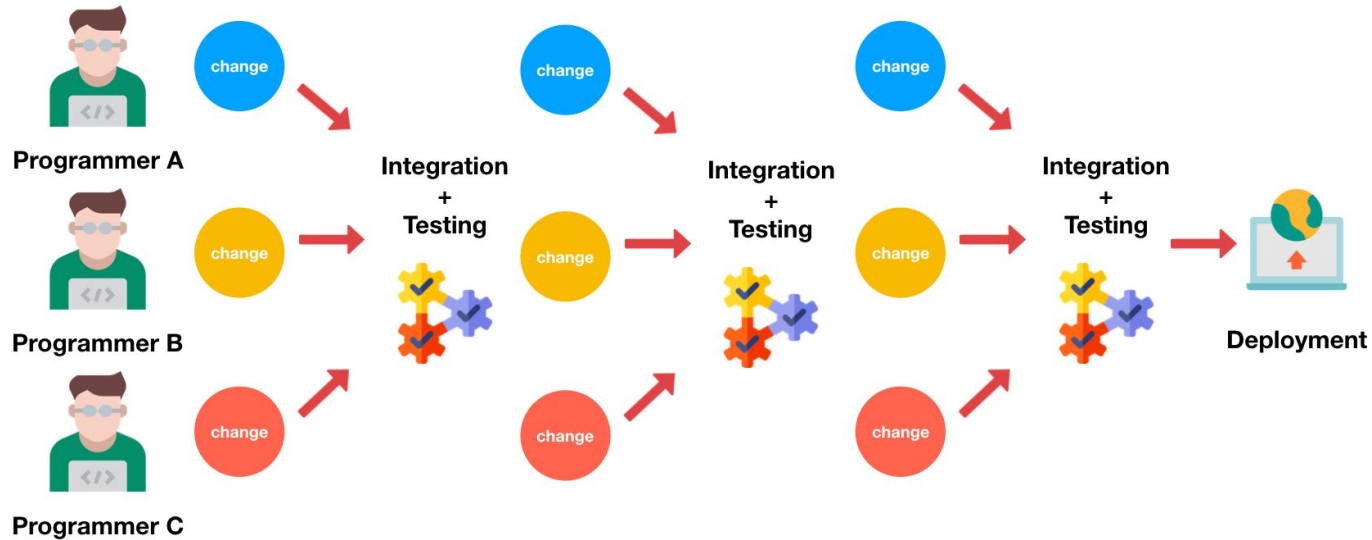
Software Development without CI/CD

Traditional Way



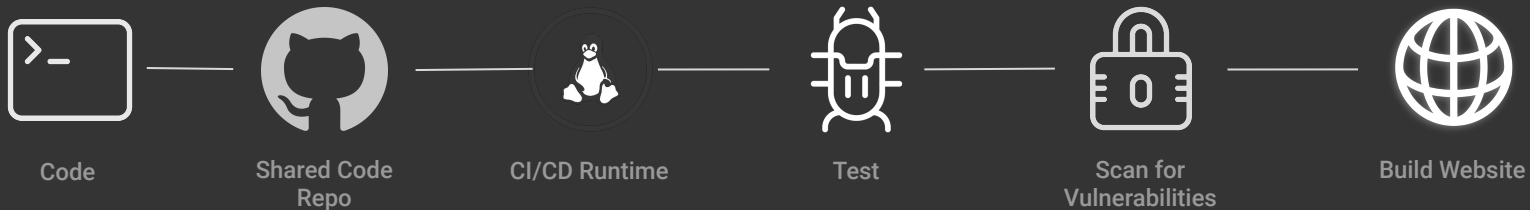
Software Development with CI/CD

With CI & CD



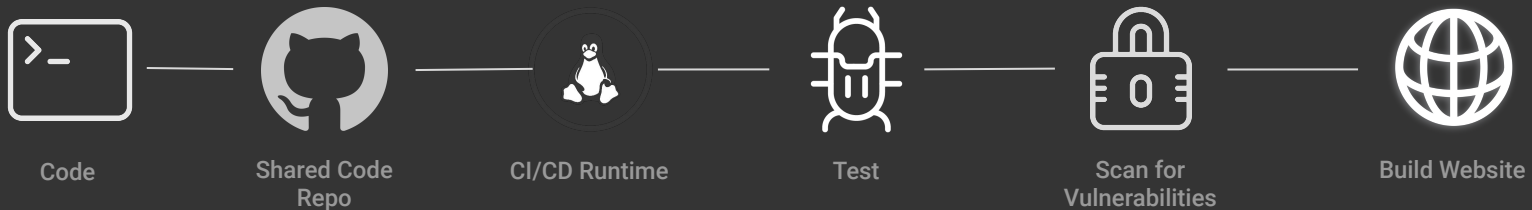
CI == Continuous Integration

the **practice** of merging all developers' working copies to a **shared** code repository



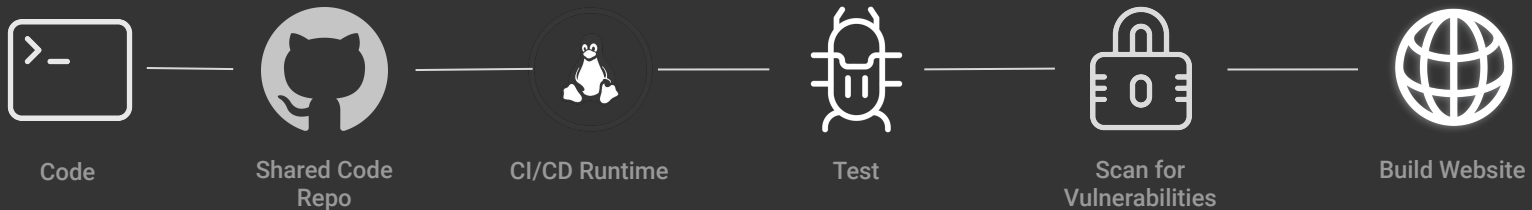
CI == Continuous Integration

- Merge code changes often



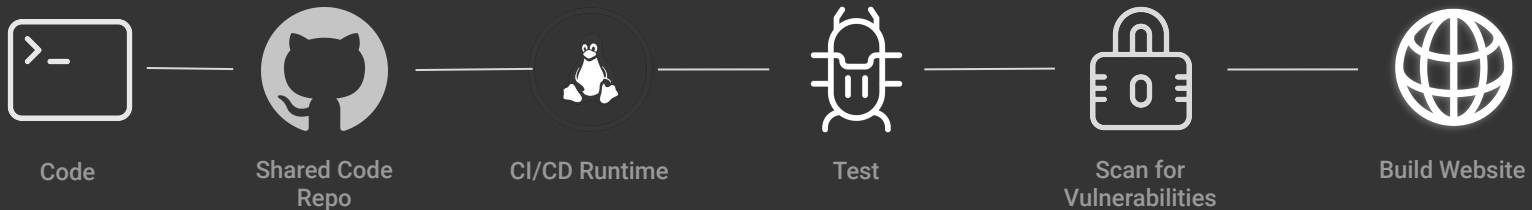
CI == Continuous Integration

- Merge code changes often
- Run automated tests to validate builds



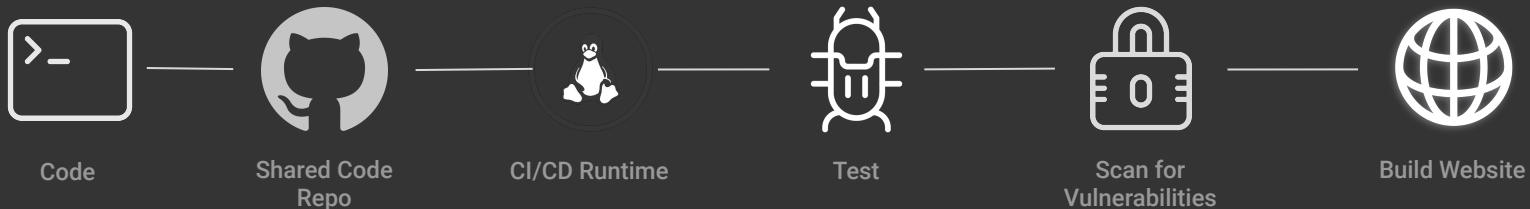
CI == Continuous Integration

- Merge code changes often
- Run automated tests to validate builds
- Only integrate tested code into code base



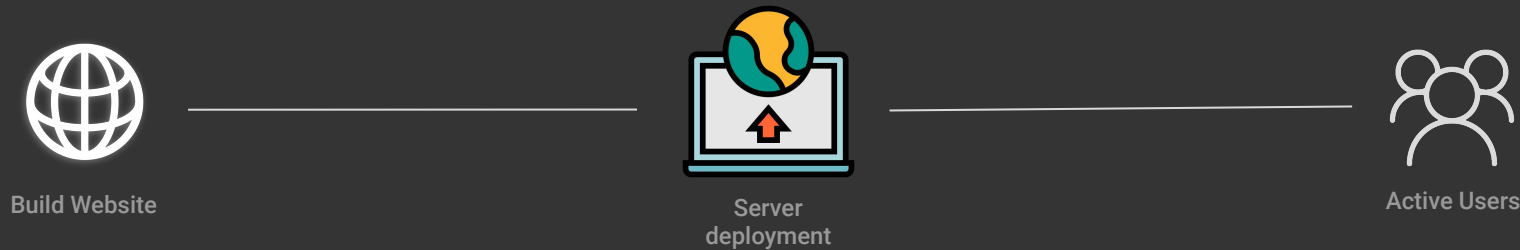
CI == Continuous Integration

- Merge code changes often
- Run automated tests to validate builds
- Only integrate tested code into code base
- Changes frequently merged into release branches



CD == Continuous Deployment

the **practice** of automatically deploying new software releases to **target environments**



Continuous Deployment

- Faster release cycles



Build Website



Server
Deployment



Active Users

Continuous Deployment

- Faster release cycles
- Low-risk releases



Build Website



Server
Deployment



Active Users

Continuous Deployment

- Faster release cycles
- Low-risk releases
- Higher quality



Build Website



Server
Deployment



Active Users

Continuous Deployment

- Faster release cycles
- Low-risk releases
- Higher quality
- Lower costs



Build Website



Server
Deployment



Active Users

Is CI/CD a TOOL?

Are there benefits to CI/CD?

CI/CD Benefits



- Improve team productivity & efficiency
- Accelerate speed to market
- Identify product/market fit
- Release higher quality, more stable products
- Increase customer satisfaction
- Keep devs happy and shipping code



I feel the need... the need for SPEED!



- Customers get access to new features faster
- Company gets a faster ROI from new features

Implementing CI/CD

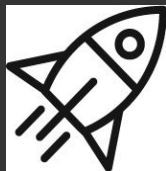
Make sure everyone is on the same page

Always start small

Do what works for you

Always measure

CI/CD Benchmarks for high performance



Throughput

At will



Duration

<10 minutes



Success Rate

> 90%



**Mean Time
to Recovery**

<1 hour

Better CI/CD Practices

- Make testing an integral part of the dev process

Better CI/CD Practices

- Make testing an integral part of the dev process
- Ensure testing environment mirrors production

Better CI/CD Practices

- Make testing an integral part of the dev process
- Ensure testing environment mirrors production
- Use coding better practices, i.e. pair programming

Better CI/CD Practices

- Make testing an integral part of the dev process
- Ensure testing environment mirrors production
- Use coding better practices, i.e. pair programming
- Automate the deploy workflow

Streamline deploy

elated-clarke-f67cc5

• Site deploy in progress

Deploys from [GitHub](#). Created at 5:48 PM.

⚙ Site settings ⚙ Domain settings

🔗 Production deploys


Production: master@HEAD **Starting Up**

5:48 PM: No deploy message

🔗 Deploy Previews

Deploy Preview #4: jerdog-patc... @a727cb0 **Building**


5:54 PM: Update config.toml



Deploy

Deployment started 9s ago...

- ▶ Fetching Source Code 6s
- ▶ Skipping Build Cache 1s
- ▶ Building
- ▶ Uploading Build Output



Preview Deployments

Deployments that are currently being worked on.

test-nextjs-vercel-qxdiffra-jerdog.ve... Preview	Queued	Update index.js jerdog-patch-1	Just now by jerdog
--	--------	-----------------------------------	--------------------



Full Report



<https://circle.ci/ssd2020>

Do not try to automate everything

Proof of concept

- A rigorous testing practice
- Consistent software environments
- Training on continuous integration practices.
- Reports to measure key metrics.

Thank you.

For feedback and swag: circle.ci/jeremy



Timeline.jerdog.me



IAmJerdog



jerdog



/in/jeremymeiss