

# Sharing Unlocking GitLab features





Somkiat Puisungnoen

Search

Somkiat | Home

Update Info 1 View Activity Log 10+ ...

Timeline About Friends 3,138 Photos More

When did you work at Opendream? X

... 22 Pending Items

Post Photo/Video Live Video Life Event

What's on your mind?

Public Post

Intro

Software Craftsmanship

Software Practitioner at สยามชานาญกิจ พ.ศ. 2556

Agile Practitioner and Technical at SPRINT3r

Somkiat Puisungnoen 15 mins · Bangkok · ...

Java and Bigdata



Facebook somkiat.cc

Page Messages Notifications 3 Insights Publishing Tools Settings Help ▾

somkiat.cc  
@somkiat.cc

Home Posts Videos Photos

Liked Following Share ... + Add a Button



**[https://github.com/up1/  
workshop-gitlab](https://github.com/up1/workshop-gitlab)**



# Topics ?

Deadline Driven Development  
Delayed project/product deliver  
Bad quality  
Low customer satisfaction





<https://about.gitlab.com/>



# What is GitLab ?

Git  
Repository

Issue  
Tracker

Merge  
Request

GitLab  
CI



# What is GitLab ?

On-premise

On-Cloud

GitLab Duo  
(AI)

Community edition

Enterprise edition



# GitLab platforms

							
Planning	Source Code Management	Continuous Integration	Security	Compliance	Artifact Registry	Continuous Delivery	Observability
DevOps Reports	Remote Development	Secrets Management	Container Scanning	Release Evidence	Virtual Registry	Release Orchestration	On-call Schedule Management
DORA Metrics	Source Code Management	Review Apps	Software Composition Analysis	Compliance Management	Container Registry	Infrastructure as Code	Incident Management
Value Stream Management	Web IDE	Code Testing and Coverage	API Security	Audit Events	Helm Chart Registry	Pages	Error Tracking
Value Stream Forecasting	GitLab CLI	Merge Trains	Coverage-guided Fuzz Testing	Software Bill of Materials	Package Registry	Feature Flags	Product Analytics Visualization
Service Desk	Code Review Workflow	Suggested Reviewers	DAST	Dependency Management	Model Registry (Beta)	Environment Management	AI Product Analytics
Wiki	Code Suggestions	Merge Request Summary	Code Quality	Vulnerability Management	Deployment Management	Auto DevOps	AI Impact Dashboard
Portfolio Management	Code Explanation	Root Cause Analysis	Secret Detection	Security Policy Management	Dependency Proxy	Metrics	Distributed Tracing
Team Planning	Code Review Summary	Discussion Summary	SAST				Logs
Generate issue description	Test Generation	Merge Commit Message Generation	Vulnerability Explanation				
Discussion Summary	Code Refactorization	Pipeline Composition and Component Catalog	Vulnerability Resolution				
Design Management	GitLab Duo for the CLI		GitLab Advisory Database				
Replacement for Jira	Replacement for GitHub	Replacement for Jenkins	Replacement for Snyk		Replacement for JFrog	Replacement for Harness	Replacement for Sentry

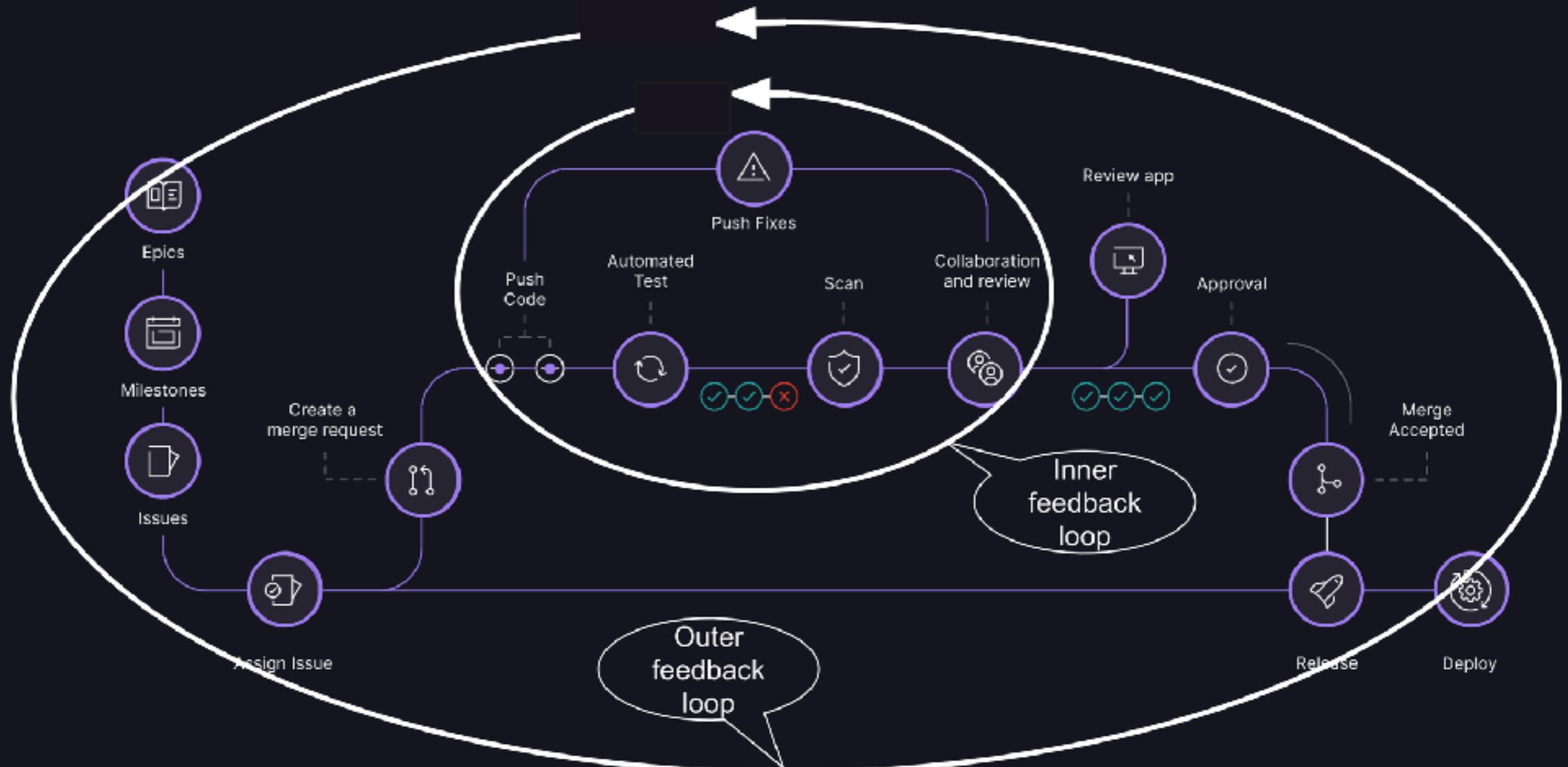
<https://about.gitlab.com/platform/>



Sharing

© 2017 - 2025 Siam Chamnankit Company Limited. All rights reserved.

# Workflow with GitLab



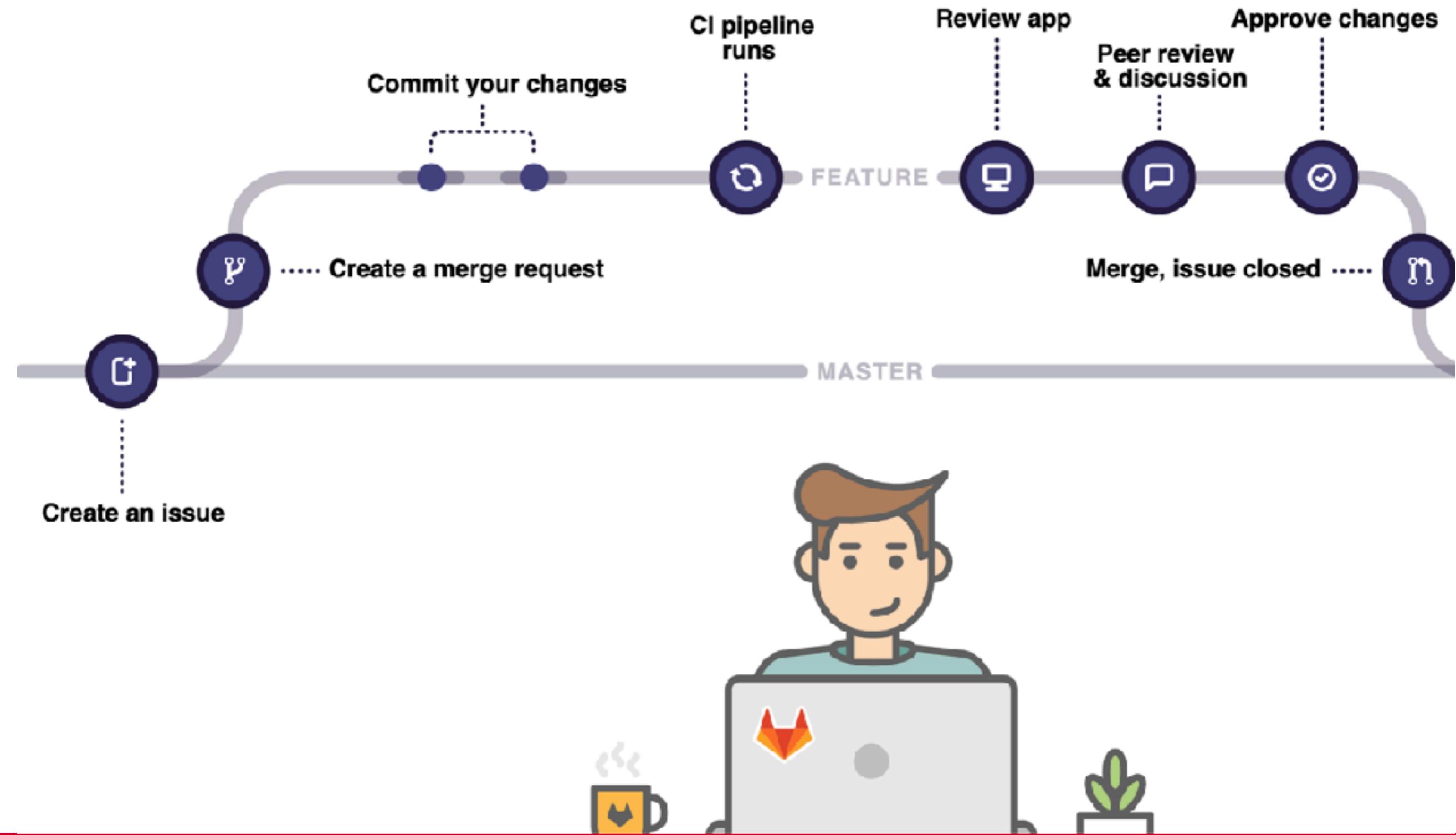
<https://about.gitlab.com/blog/2023/07/27/gitlab-flow-duo/>



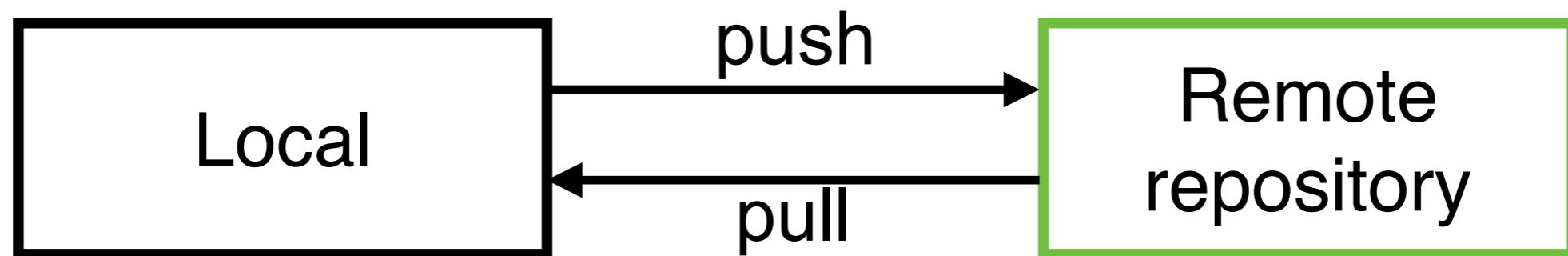
Sharing

© 2017 - 2025 Siam Chamnkit Company Limited. All rights reserved.

# Workflow with GitLab



# Git repository with GitLab



GitLab CLI



Bitbucket



# Branch Strategies

Trunk-Based Development (TBD)

  Feature branch

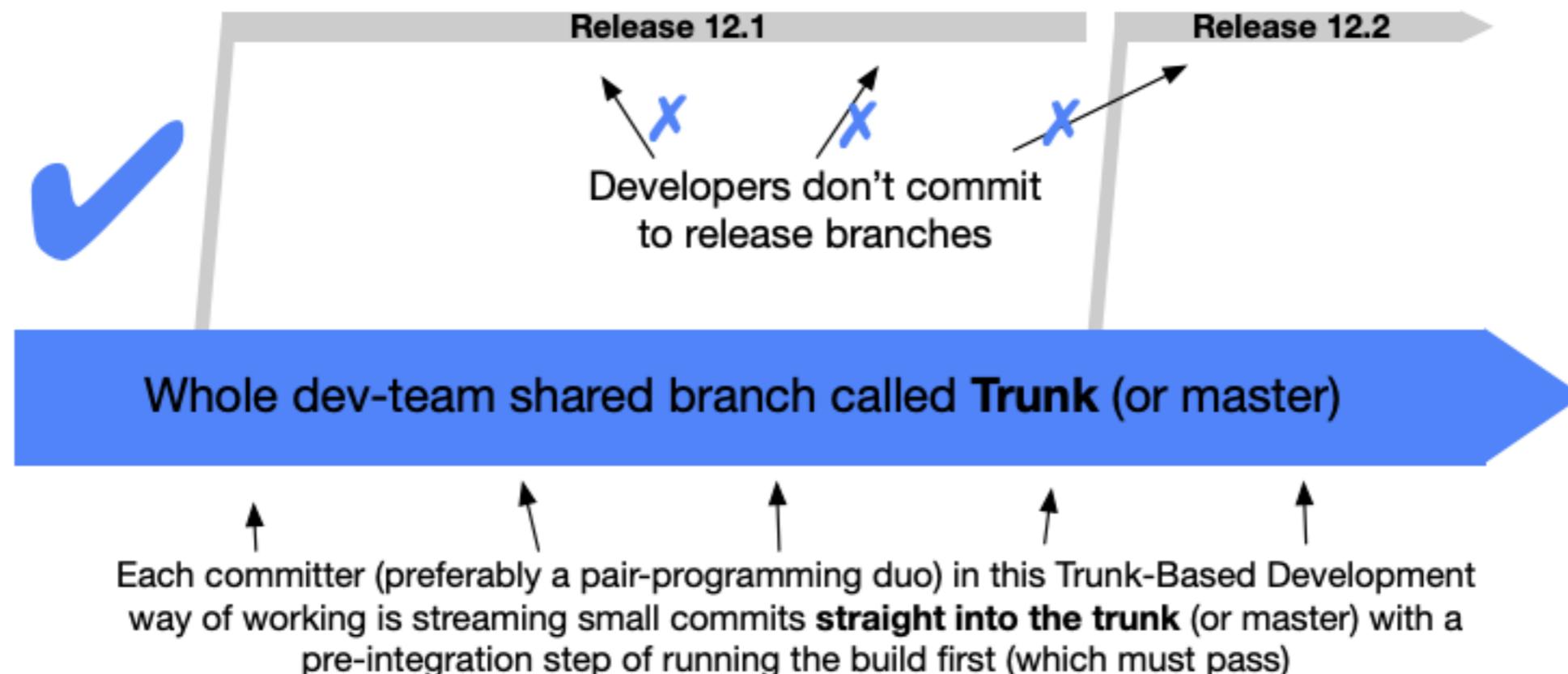
  Environment branch

...  
...



# Trunk-Based Development

For small team



<https://trunkbaseddevelopment.com/>

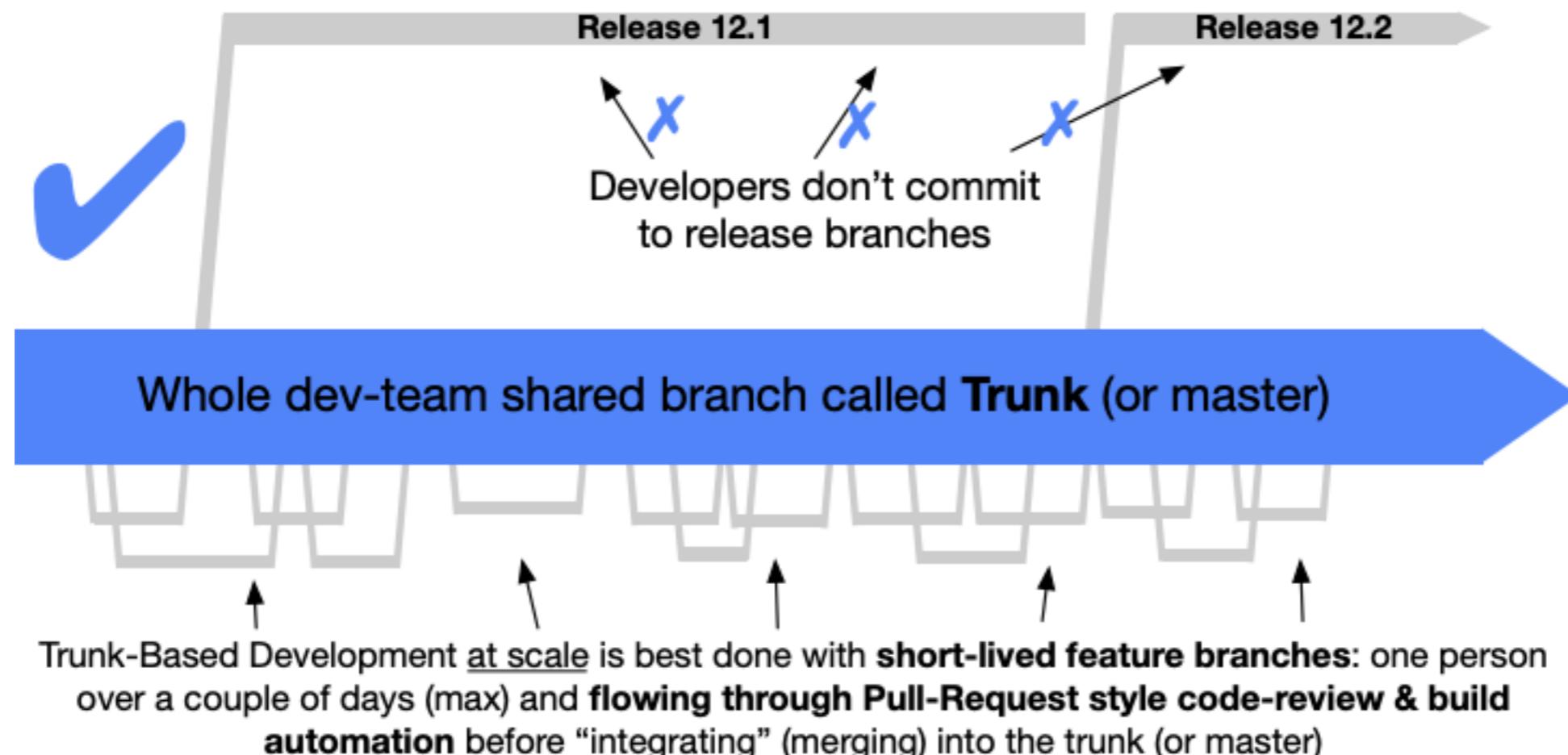


Sharing

© 2017 - 2025 Siam Chamnankit Company Limited. All rights reserved.

# Trunk-Based Development

## Scale TBD with short-lived feature branch



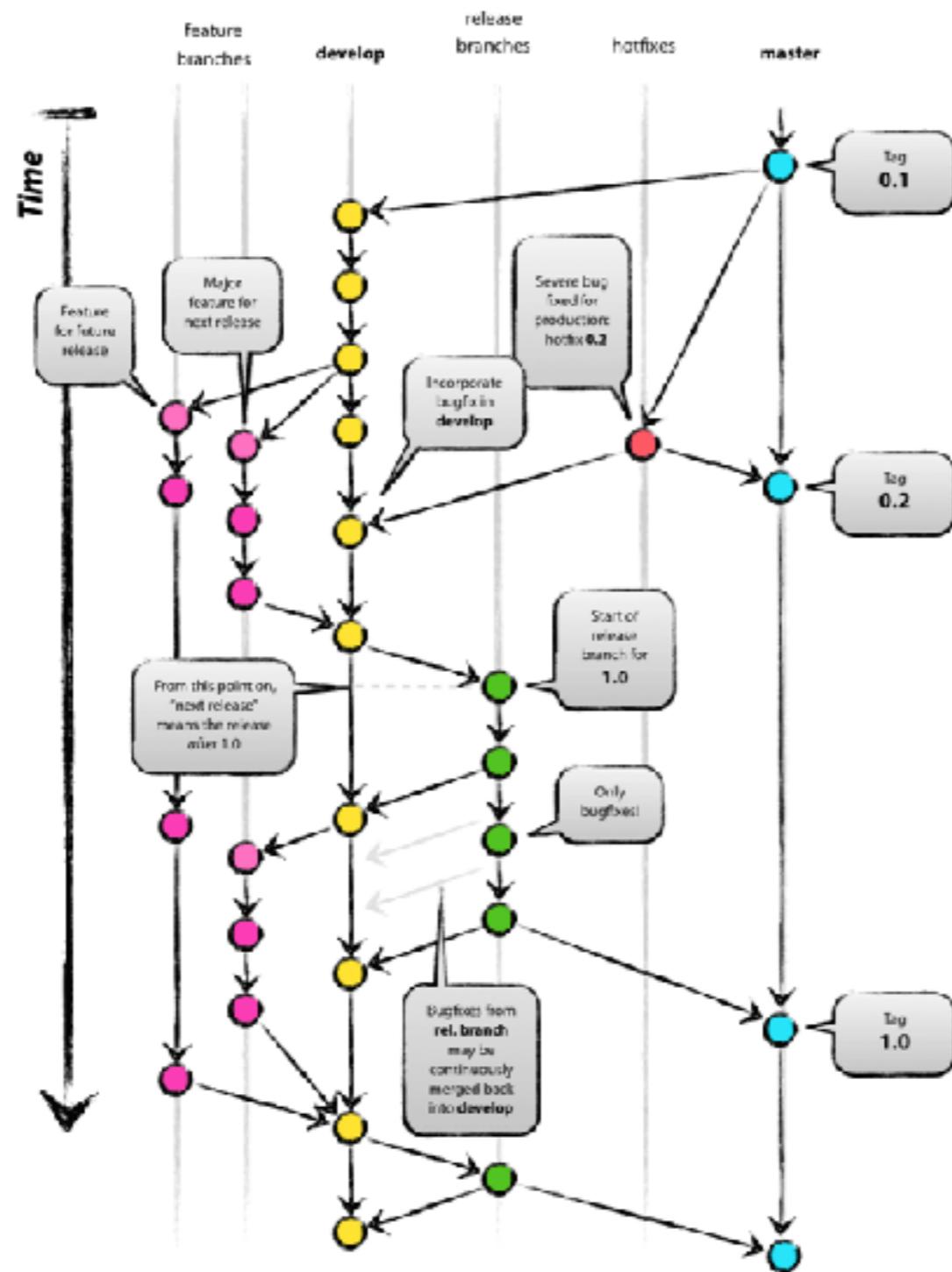
<https://trunkbaseddevelopment.com/>



Sharing

© 2017 - 2025 Siam Chamnkit Company Limited. All rights reserved.

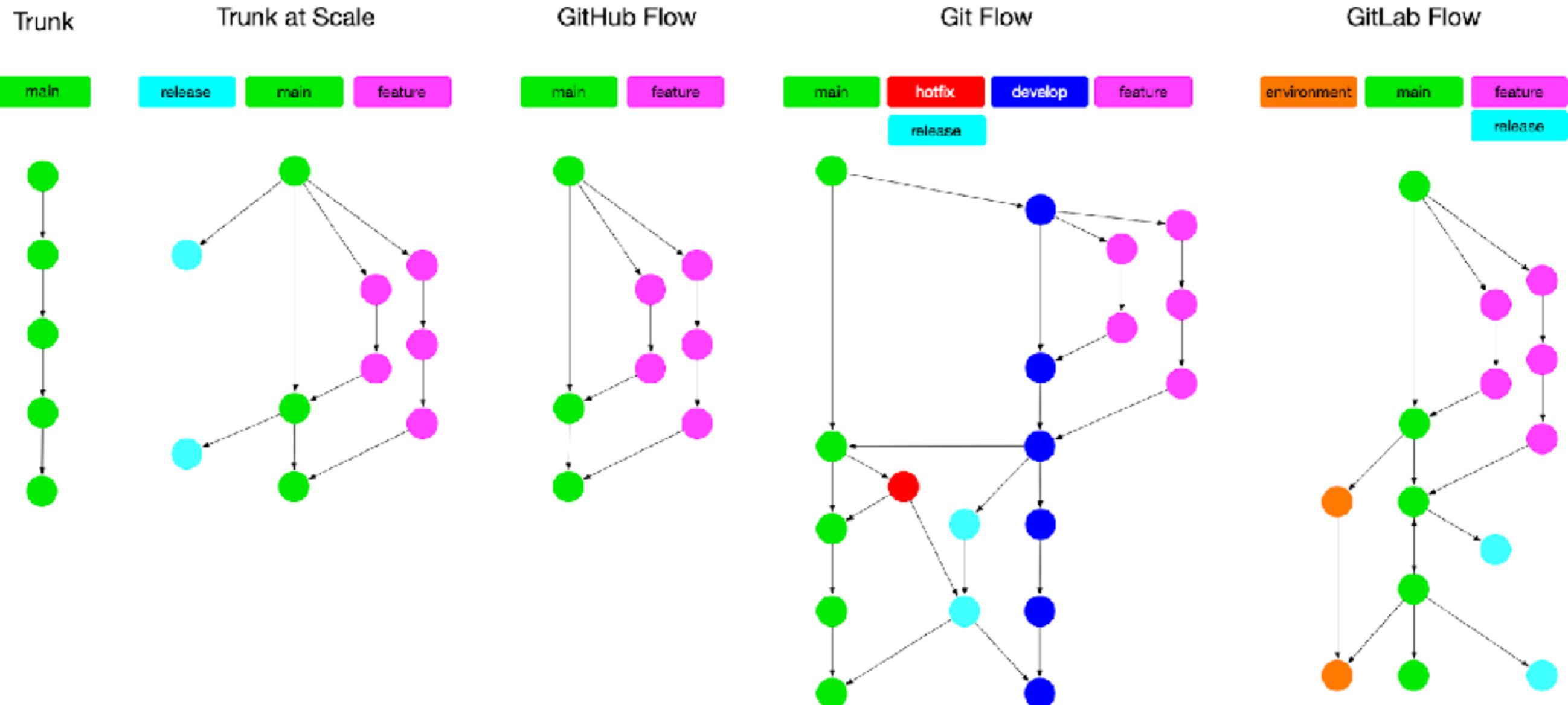
# Git Branching Model



<https://nvie.com/posts/a-successful-git-branching-model/>



# Git Branching Models !!



<https://articles.merapar.com/hello-serverless-goodbye-git-flow>

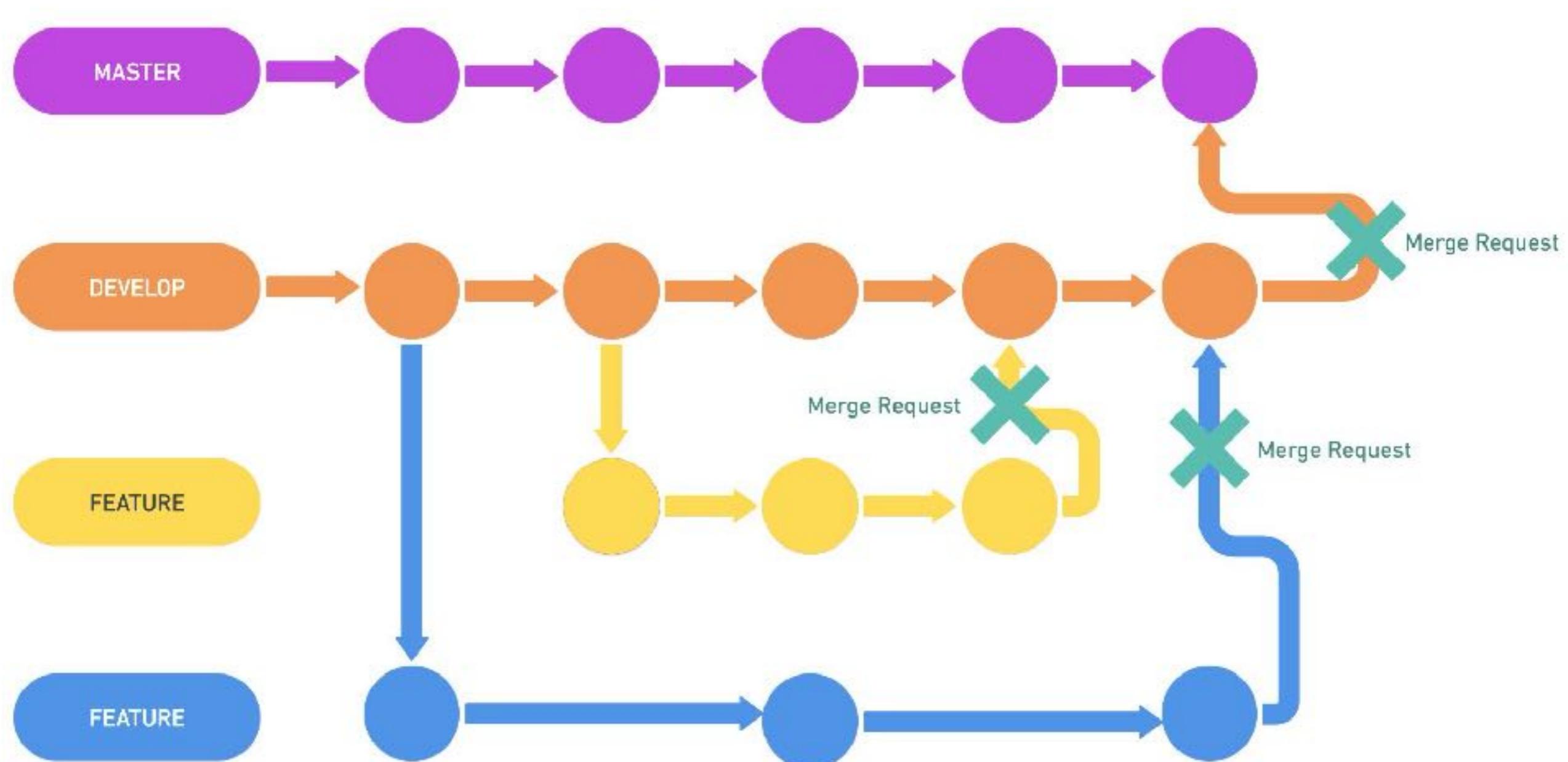


# Merge Request (MR)

[https://docs.gitlab.com/ee/user/project/merge\\_requests/](https://docs.gitlab.com/ee/user/project/merge_requests/)



# Merge Request (MR)

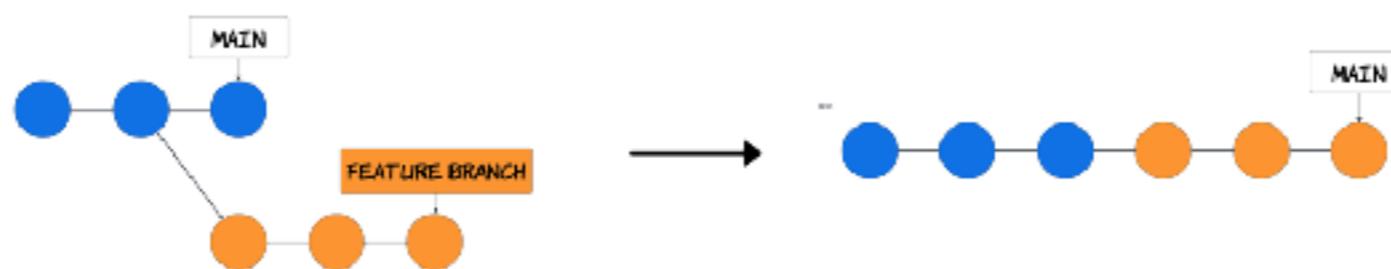


# Merge strategies ?

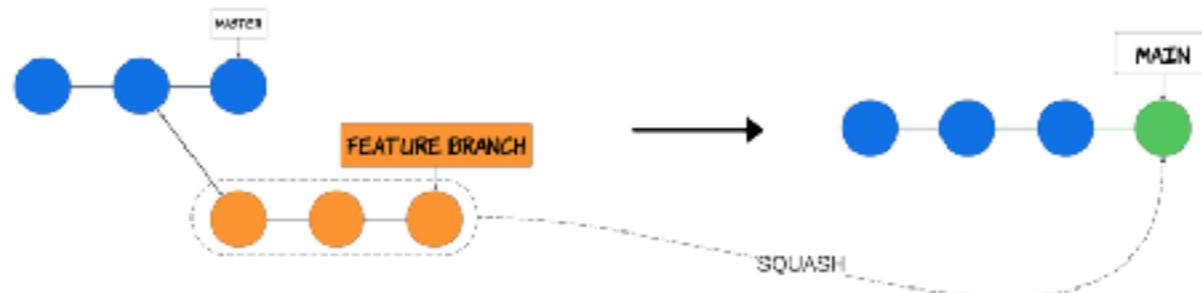
MERGE



REBASE



SQUASH MERGE



# Merge in GitLab

Goto project settings => Merge request

## Merge requests

Choose your merge method, merge options, merge checks, and merge suggestions.

Additional settings that influence how and when merges are done.

- Automatically resolve merge request diff threads when they become outdated
- Show link to create or view a merge request when pushing from the command line
- Enable "Delete source branch" option by default  
Existing merge requests and protected branches are not affected.

### Squash commits when merging

Set the default behavior of this option in merge requests. Changes to this are also applied to existing merge requests. [What is squashing?](#)

- Do not allow  
Squashing is never performed and the checkbox is hidden.
- Allow  
Checkbox is visible and unselected by default.
- Encourage  
Checkbox is visible and selected by default.
- Require  
Squashing is always performed. Checkbox is visible and selected, and users cannot change it.



# Merge commit message

Goto project settings => Merge request

## Merge commit message template

The commit message used when merging, if the merge method creates a merge commit.

Merge branch '{source\_branch}' into '{target\_branch}'

{title}

{issues}

See merge request {reference}

Leave empty to use default template. Maximum 500 characters. [What variables can I use?](#)

## Squash commit message template

The commit message used when squashing commits.

{title}

Leave empty to use default template. Maximum 500 characters. [What variables can I use?](#)

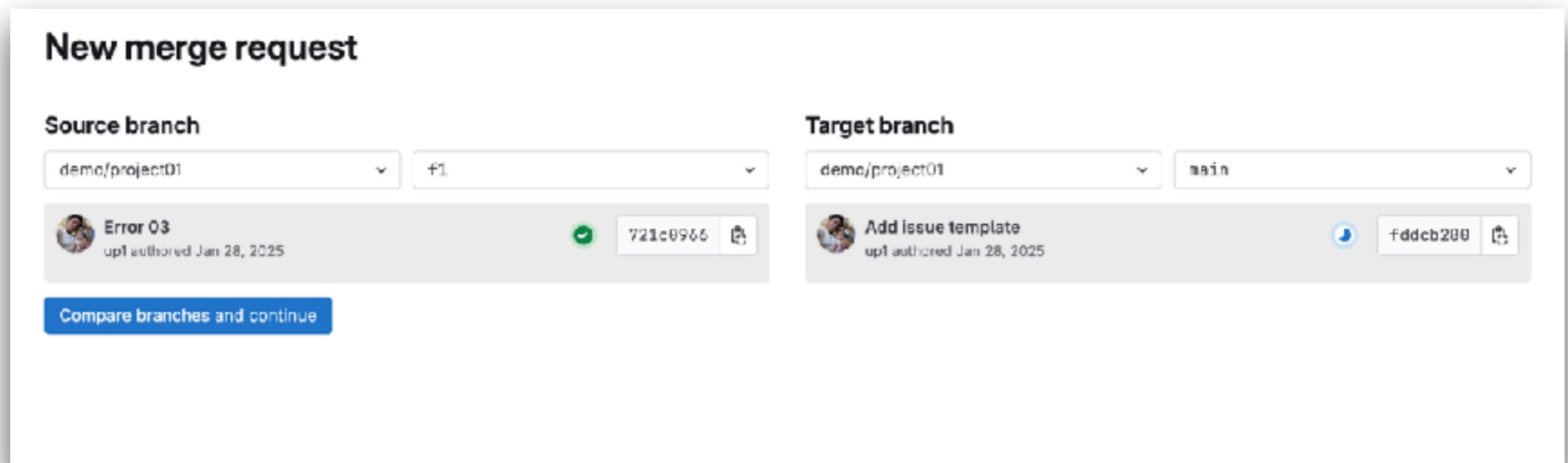


Sharing

© 2017 - 2025 Siam Chamnkit Company Limited. All rights reserved.

# Create merge request in UI (1)

Goto project settings => Code => Merge request



# Create merge request in UI (2)

## Create merge request in Issue

The screenshot shows a Jira issue page for an incident titled "Error 01". At the top right, there are "Edit" and three-dot buttons. Below the title, there are "Open" and "Incident created 13 hours ago by Administrator" status indicators. A navigation bar has "Summary" underlined and "Timeline". The main content area starts with the incident title "Error 01". Below it are "Like" (0), "Dislike" (0), and "Comment" buttons. To the right is a blue "Create merge request" button with a dropdown arrow. Underneath is a "Child items" section with an "Add" button and a note: "No child items are currently assigned. Use child items to break down this issue into smaller parts." Below that is a "Linked incidents or issues" section with an "Add" button and a note: "Link incidents together to show that they're related. [Learn more](#)". At the bottom is an "Activity" section with a "Sort or filter" button and two items: "Administrator added critical label 13 hours ago" and "Administrator mentioned in commit ff9c6dc7 13 hours ago".



# Create merge request in CLI

Use Git CLI

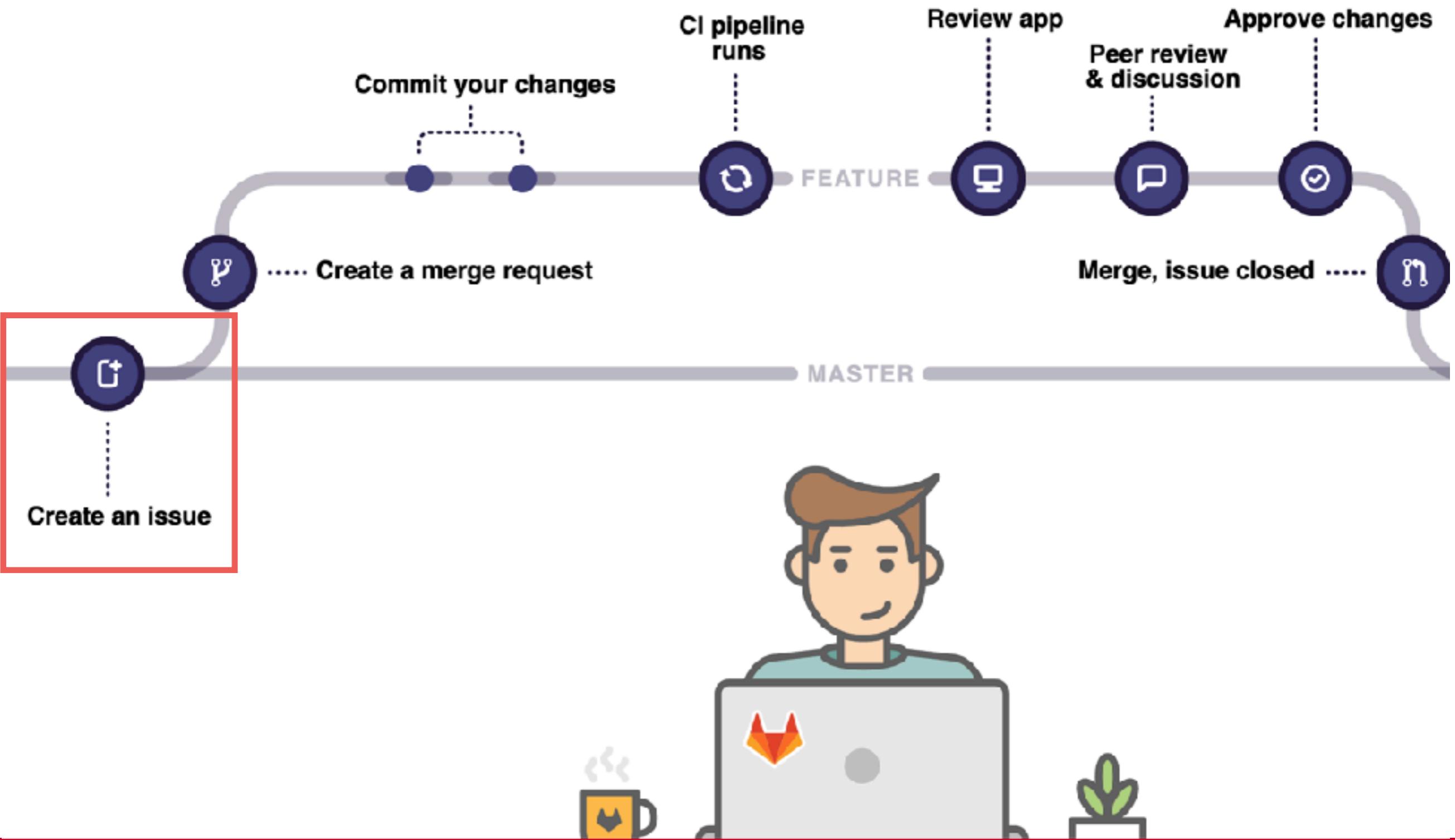
```
git push -o merge_request.create origin feature-branch  
git push -o merge_request.title=""  
git push -o merge_request.description=""  
  
git push -o merge_request.remove_source_branch
```



# **Start with an Issue !!**



# Workflow with GitLab



# Issue Tracker

Discussion  
Planing  
Assignment  
Organize



[https://docs.gitlab.com/ee/user/project/issue\\_board.html](https://docs.gitlab.com/ee/user/project/issue_board.html)



# Issue Board

Create -> Manage -> Workflow -> Milestone

The screenshot shows the GitLab Issue Board interface with four columns:

- Backlog:** 11673 issues. Includes items like "Improve consistency in the way we retrieve project & group in API endpoints" (#20728) and "Wiki Page History appears to direct to wrong link and 404s" (#20528).
- In dev:** 33 issues. Includes items like "JUnit XML Test Summary in MR widget" (#45318) and "Multi JIRA issue transition allows" (#40002).
- In review:** 24 issues. Includes items like "'ExploreBuildArtifactsWorker' is broken" (#41057) and "Ensure that all CI/CD queries take less than 15 seconds to complete" (#40028).
- Closed:** 23542 issues. Includes items like "The activity feed is not accessible for empty projects" (#29577) and "Sequential scans on 'routes' table increased from 0 to 1 billion scans per minute" (#20488).



# Issue Board with progress !!

Progress

216 issues: 186 open and 161 closed (total weight: 107) 46% complete 14 days remaining

+ New Issue    Browse Issues

Issues 216    Merge Requests 131    Participants 24    Labels 47

Unstarted Issues (open and unassigned)	Ongoing Issues (open and assigned)	Completed Issues (closed)
Links in activity feed and todos need a title attribute #14042 Frontend up-for-grabs	Reuse existing method to force caching in badges controller #14100 CI	Automatic issue sidebar collapse should not persist #14037 Frontend
Can't copy markdown snippets #13882 Frontend bug regression up-for-grabs	Include mentions in commits page to ToDos #14006 Core feature proposal todos	Add some space between flash message and login screen #13935 Frontend



# Create Issue with template

Go to projects => Issue

New Issue

Title (required)

Type [?](#)

Issue

Description

Choose a template

Choose a template

Normal text  **B** *I*

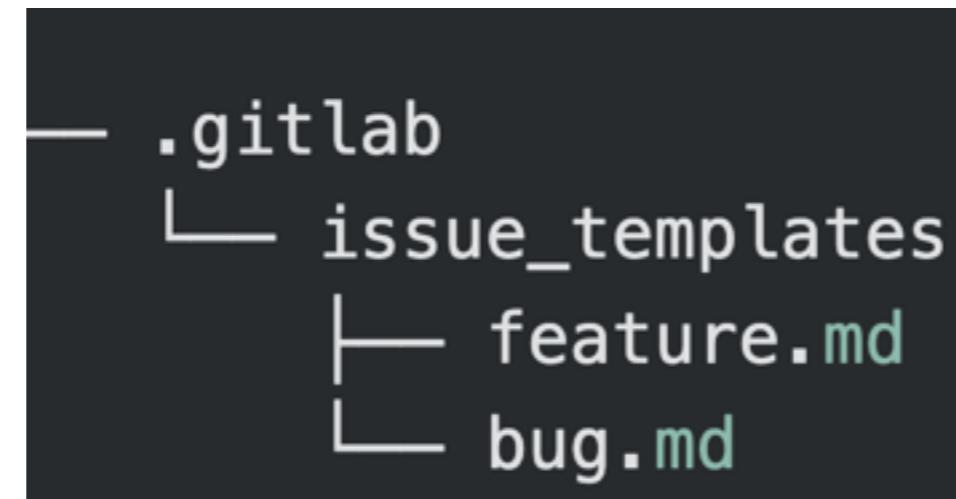
Switch to plain text editing

This issue is confidential and should only be visible to team members with at least the Planner role.



# Issue Template in GitLab

Create file in ***.gitlab/issue\_templates***  
Markdown format



# Label name ?

Categorize and filter issues and merge requests

The screenshot shows a list of labels in a GitLab interface. Each label entry includes the label name, a small preview icon, the project it belongs to, and links for Issues, Merge requests, Subscribe, and more options.

Label	Project	Issues	Merge requests	Subscribe	More
bug	demo / project01				
confirmed	demo / project01				
critical	demo / project01				
discussion	demo / project01				
documentation	demo / project01				
enhancement	demo / project01				
suggestion	demo / project01				
support	demo / project01				

<https://docs.gitlab.com/ee/user/project/labels.html>



Sharing

© 2017 - 2025 Siam Chamnankit Company Limited. All rights reserved.

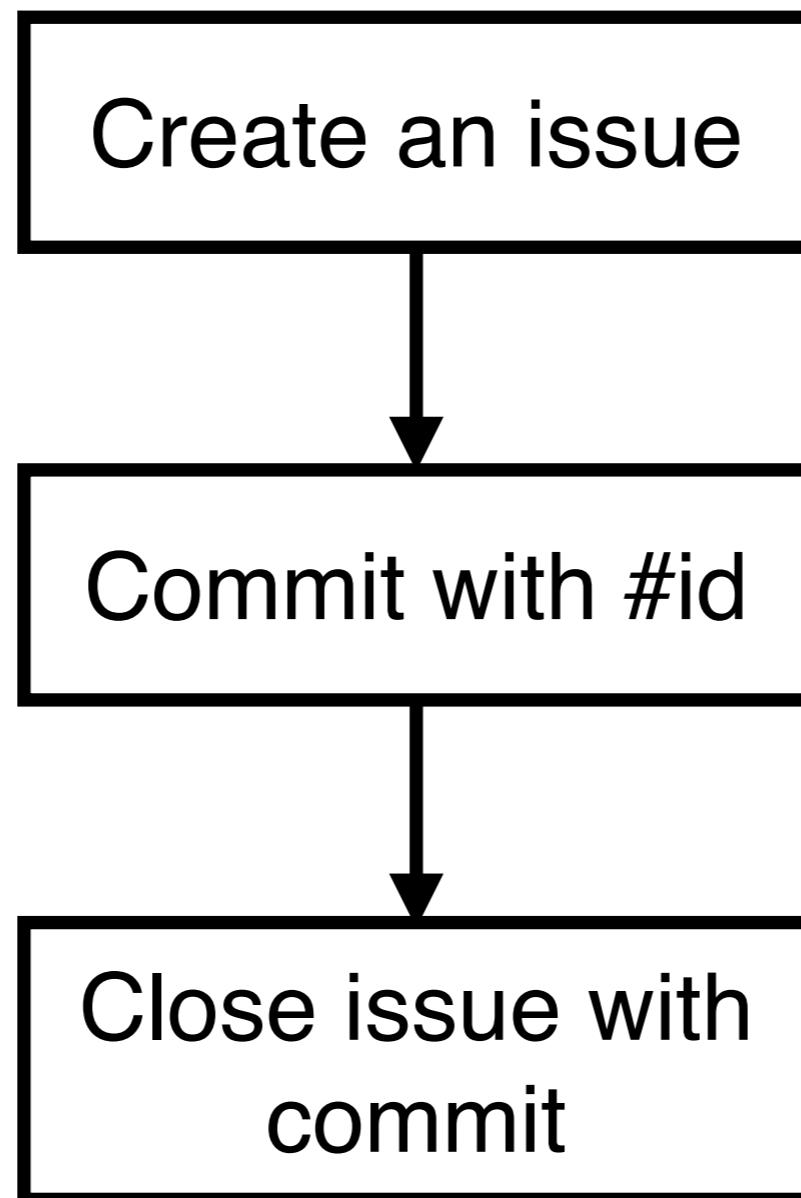
# Update status of Issue ?

Manual

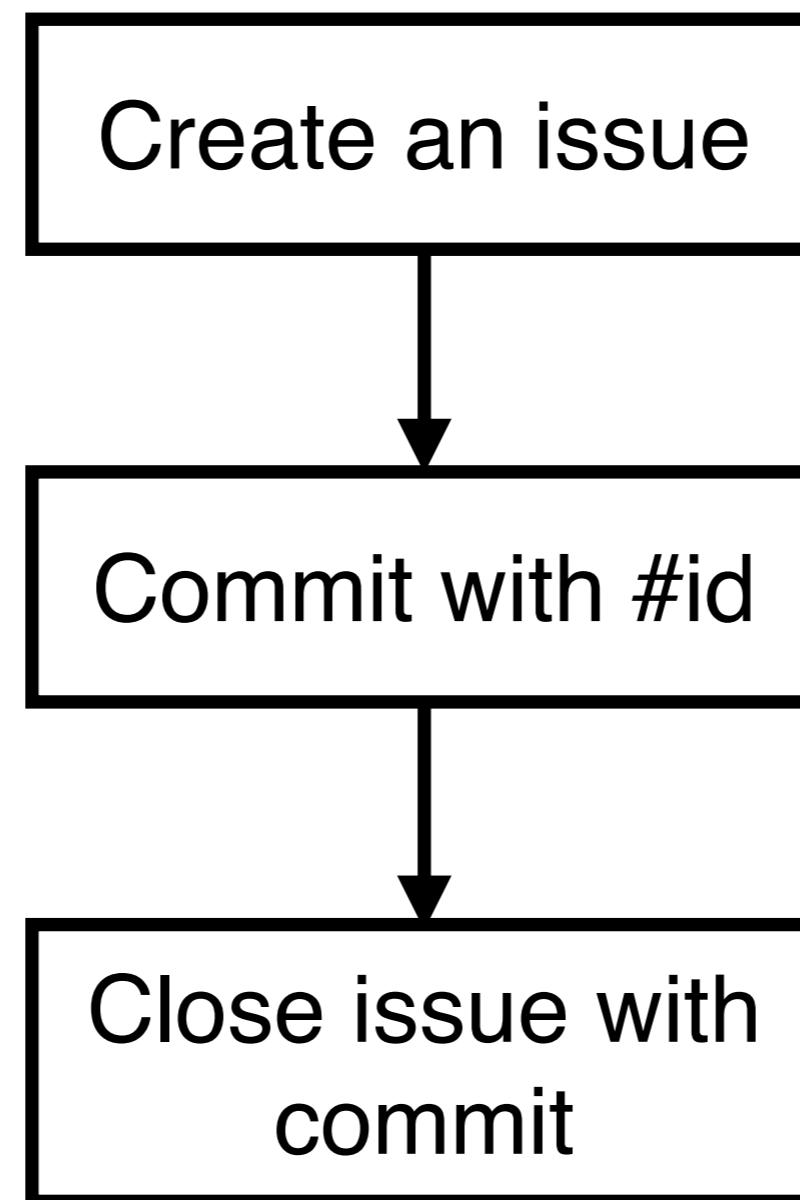
Commit  
message



# Workflow (1)



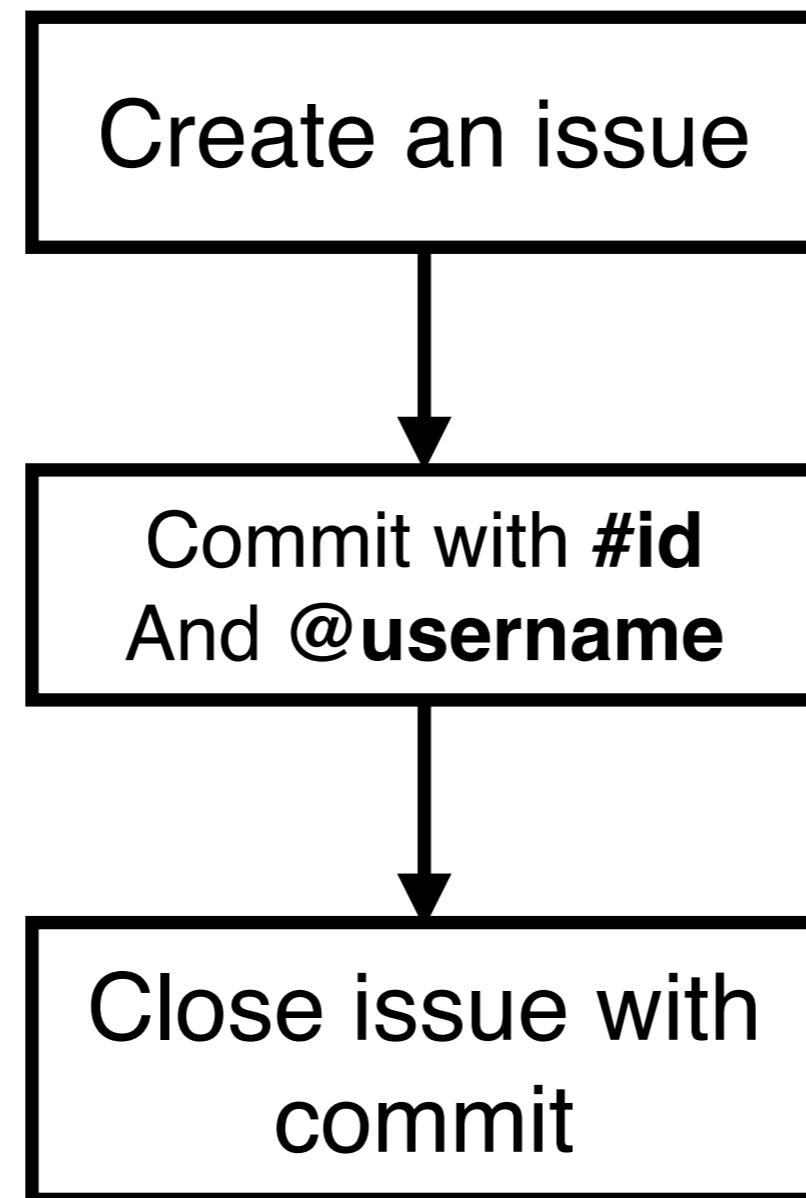
# Workflow (2)



Issue template ?  
Label ?



# Workflow (3)

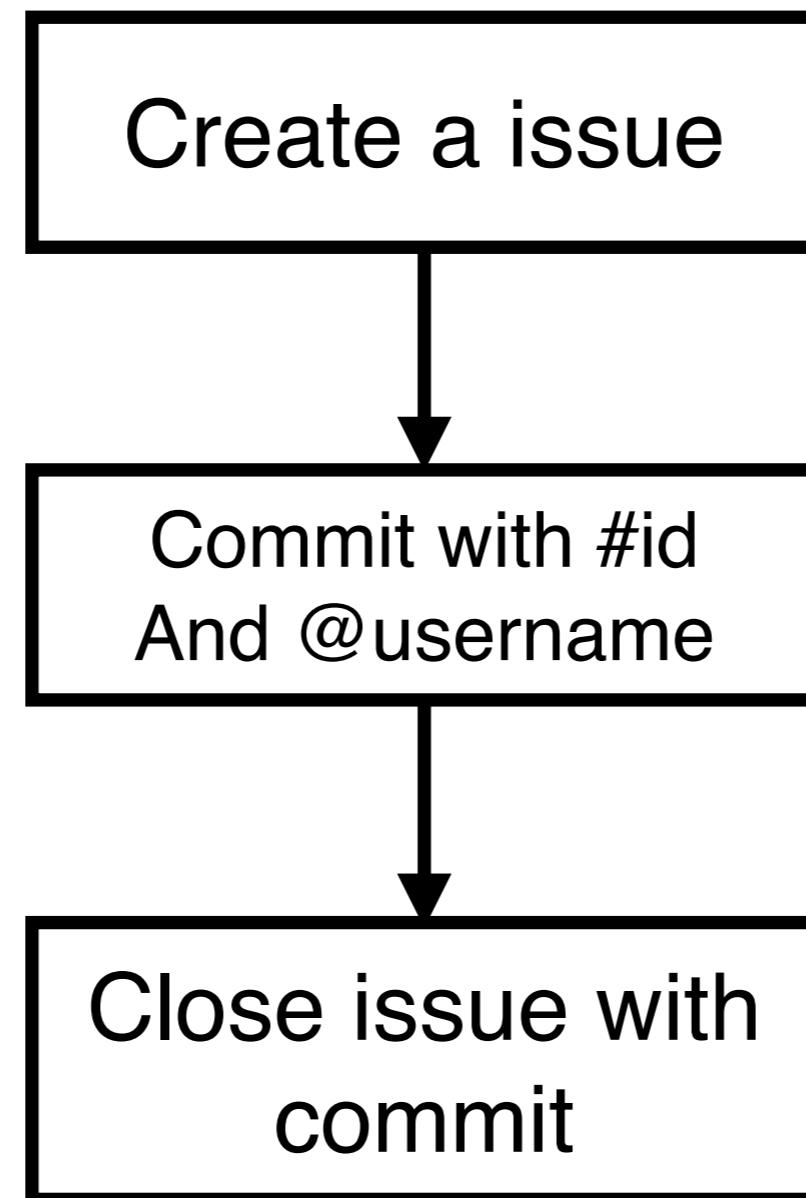


Issue template ?  
Label ?

Reference to issue id  
Assign to user



# Workflow (4)



Issue template ?  
Label ?

Reference to issue id  
Assign to user

Closing patterns  
eg. Closed, Fixed

[https://docs.gitlab.com/ee/user/project/issues/managing\\_issues.html#default-closing-pattern](https://docs.gitlab.com/ee/user/project/issues/managing_issues.html#default-closing-pattern)



Sharing

© 2017 - 2025 Siam Chamnankit Company Limited. All rights reserved.

# Good Commit message ?

	COMMENT	DATE
○	CREATED MAIN LOOP & TIMING CONTROL.	14 HOURS AGO
○	ENABLED CONFIG FILE PARSING	9 HOURS AGO
○	MISC BUGFIXES	5 HOURS AGO
○	CODE ADDITIONS/EDITS	4 HOURS AGO
○	MORE CODE	4 HOURS AGO
○	HERE HAVE CODE	4 HOURS AGO
○	AAAAAAA	3 HOURS AGO
○	ADKFJ5LKDFJSOKLFJ	3 HOURS AGO
○	MY HANDS ARE TYPING WORDS	2 HOURS AGO
○	HAAAAAAAAANDS	2 HOURS AGO

AS A PROJECT DRAGS ON, MY GIT COMMIT MESSAGES GET LESS AND LESS INFORMATIVE.

- bug fix
- update readme.md
- made changes
- added files
- wrote something to make it work



# Good Commit message ?

```
daccff1f test should pass
3fff19f6 test should pass
5b998d9a add a disabled property for the button
06faab4d fix lint
186cce90 refactor button
4b99d91a fix spinner component
5b998d9a fix css
263288a5 test should pass
c3fb85af Create Button component
```



# Better commit message

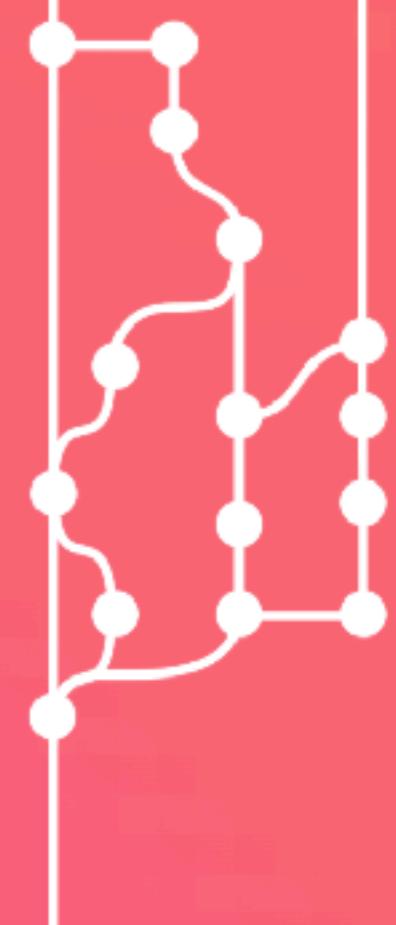
## Conventional Commits

A specification for adding human and machine readable meaning to commit messages

Quick Summary

Full Specification

Contribute



<https://www.conventionalcommits.org/en/v1.0.0/>



# Better commit message

## Commit message with description and breaking change footer

```
feat: allow provided config object to extend other configs
```

```
BREAKING CHANGE: `extends` key in config file is now used for extending other config files
```

## Commit message with ! to draw attention to breaking change

```
feat!: send an email to the customer when a product is shipped
```

## Commit message with scope and ! to draw attention to breaking change

```
feat(api)!: send an email to the customer when a product is shipped
```

<https://www.conventionalcommits.org/en/v1.0.0/>



# CommitLint

The screenshot shows the homepage of the CommitLint website. At the top left is the CommitLint logo (a stylized 'cl' icon). Next to it is the word "commitlint". To the right is a search bar with a magnifying glass icon and a key icon. On the far right are navigation links: "Home", "Guides", and "Reference". Below the header, the main title "commitlint" is in blue, followed by "Lint commit messages" in large, bold, dark gray text. A subtitle "helps your team adhere to a commit convention" is in smaller dark gray text. To the right of the main text area is a dark rectangular box containing a terminal session output:

```
->/Projects/labs/commitlint master
> echo "foo" | commitlint
x  input: foo
•  message may not be empty [subject-empty]
x  type may not be empty [type-empty]
x  Found 2 problems, 0 warnings
->/Projects/labs/commitlint master
>
```

<https://commitlint.js.org/>



# CommitLint and Husky

```
→ project01 git:(main) ✘ git commit -m "Add demo.txt"
✗ input: Add demo.txt
✖ Please add rules to your `commitlint.config.js`
  - Getting started guide: https://commitlint.js.org/guides/getting-started/
  - Example config: https://github.com/conventional-changelog/commitlint/b
%40commitlint/config-conventional/src/index.ts [empty-rules]

✖ found 1 problems, 0 warnings
ⓘ Get help: https://github.com/conventional-changelog/commitlint/#what-is-
husky - commit-msg script failed (code 9)
```

<https://typicode.github.io/husky/>



Sharing

© 2017 - 2025 Siam Chamnkit Company Limited. All rights reserved.

# VSCode Conventional Commit

The screenshot shows the Visual Studio Marketplace page for the 'Conventional Commits' extension. At the top, there's a navigation bar with the Visual Studio logo and 'Marketplace'. Below it, a breadcrumb navigation shows 'Visual Studio Code > Snippets > Conventional Commits'. The main content area features a large circular icon on the left, which is circled in red in the original image. To the right of the icon, the extension name 'Conventional Commits' is displayed in bold. Below the name, the author 'vivaxy' is listed, along with the number of installs '485,681' and a five-star rating '(36)'. A 'Free' label is also present. A description below the rating reads 'Conventional Commits for VSCode.' At the bottom of the card, there are two buttons: a green 'Install' button and a link 'Trouble Installing?'. The entire screenshot is set against a light gray background.

<https://marketplace.visualstudio.com/items?itemName=vivaxy.vscode-conventional-commits>



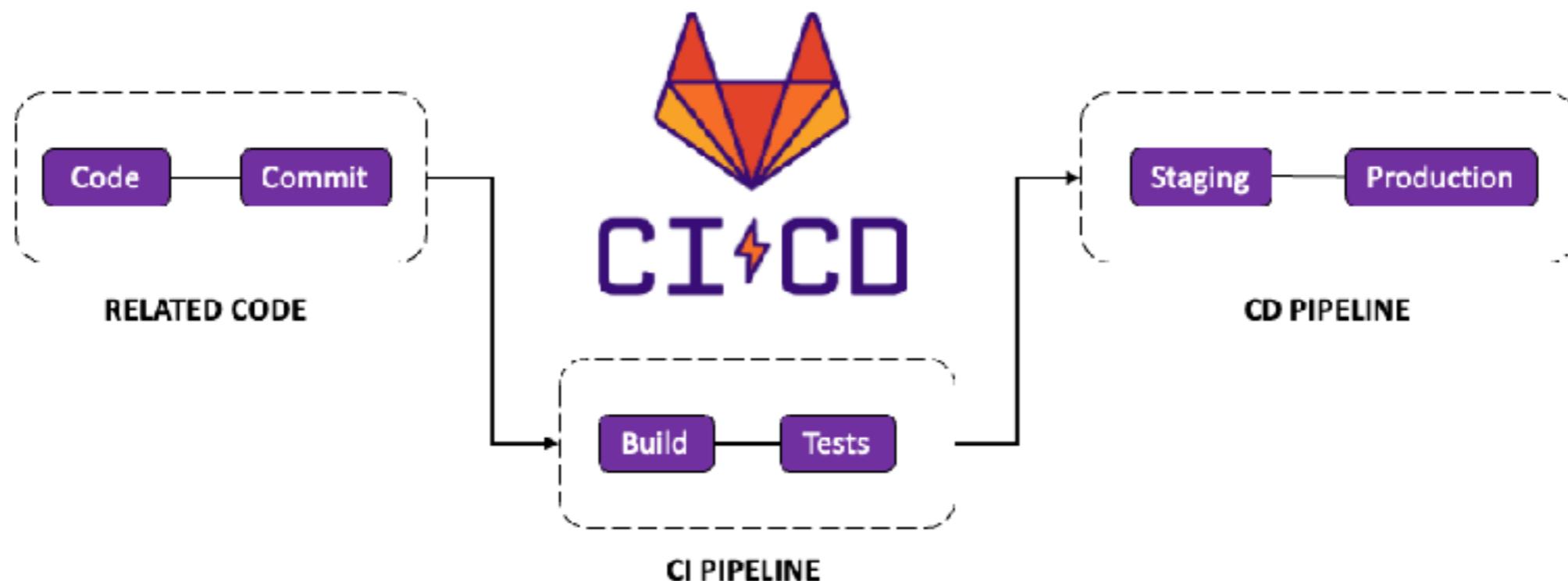
Sharing

© 2017 - 2025 Siam Chamnkit Company Limited. All rights reserved.

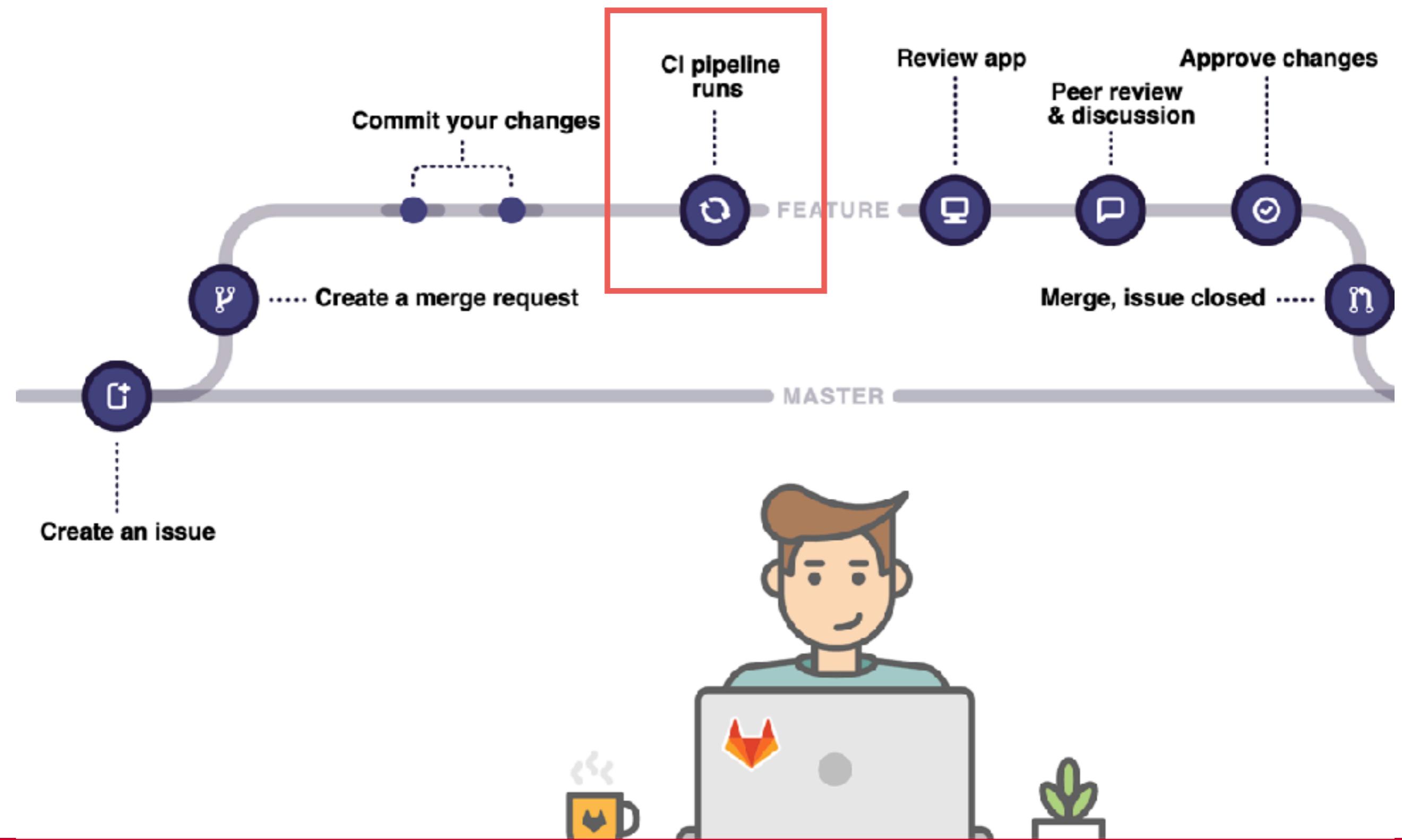
# Demo



# CI/CD in GitLab



# Workflow with GitLab



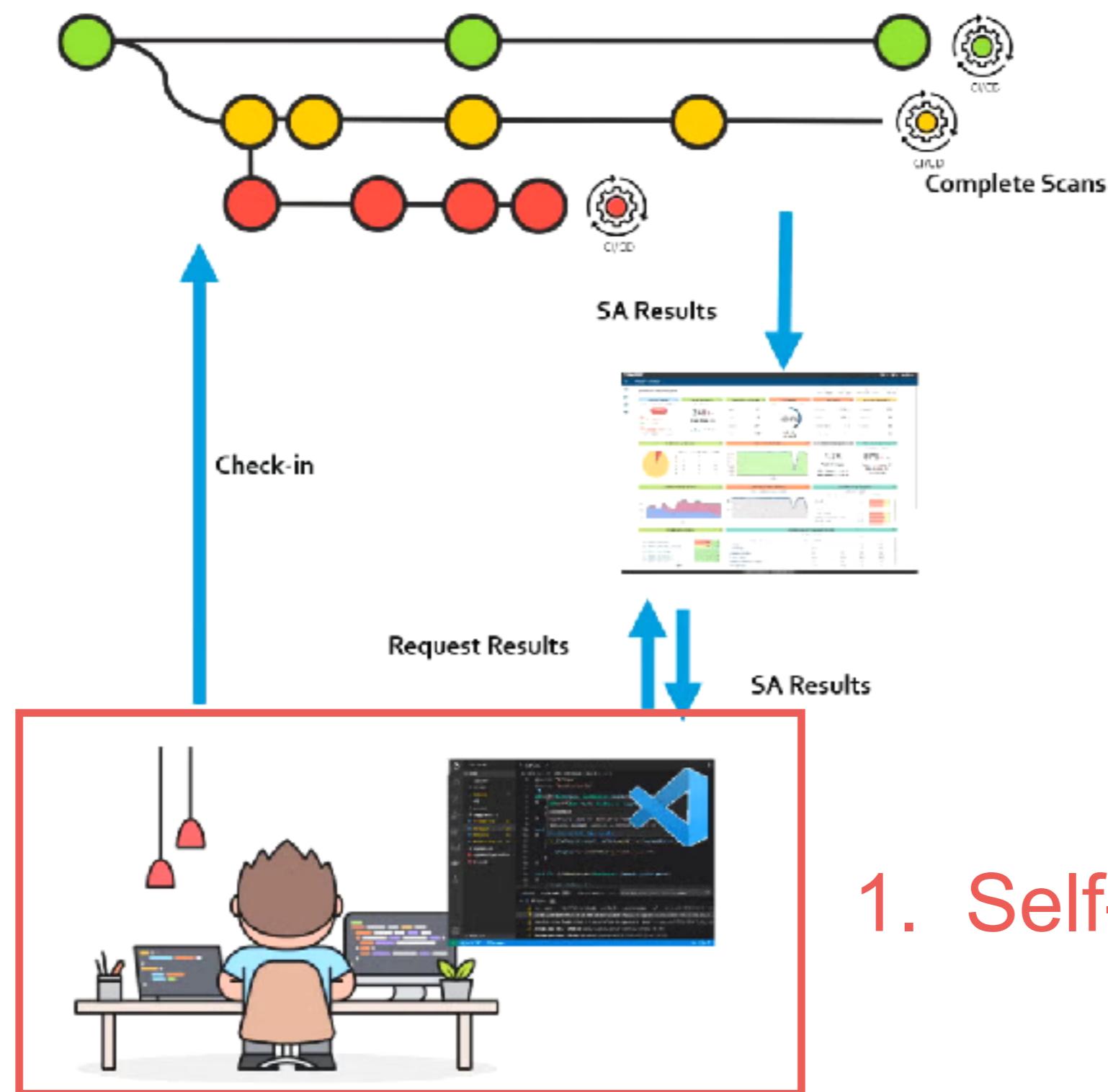
**Continuous Integration  
is a software development practice  
where each member of a team merges  
their changes into a codebase together with  
their colleagues changes at least daily.**

*--- Martin Fowler ---*

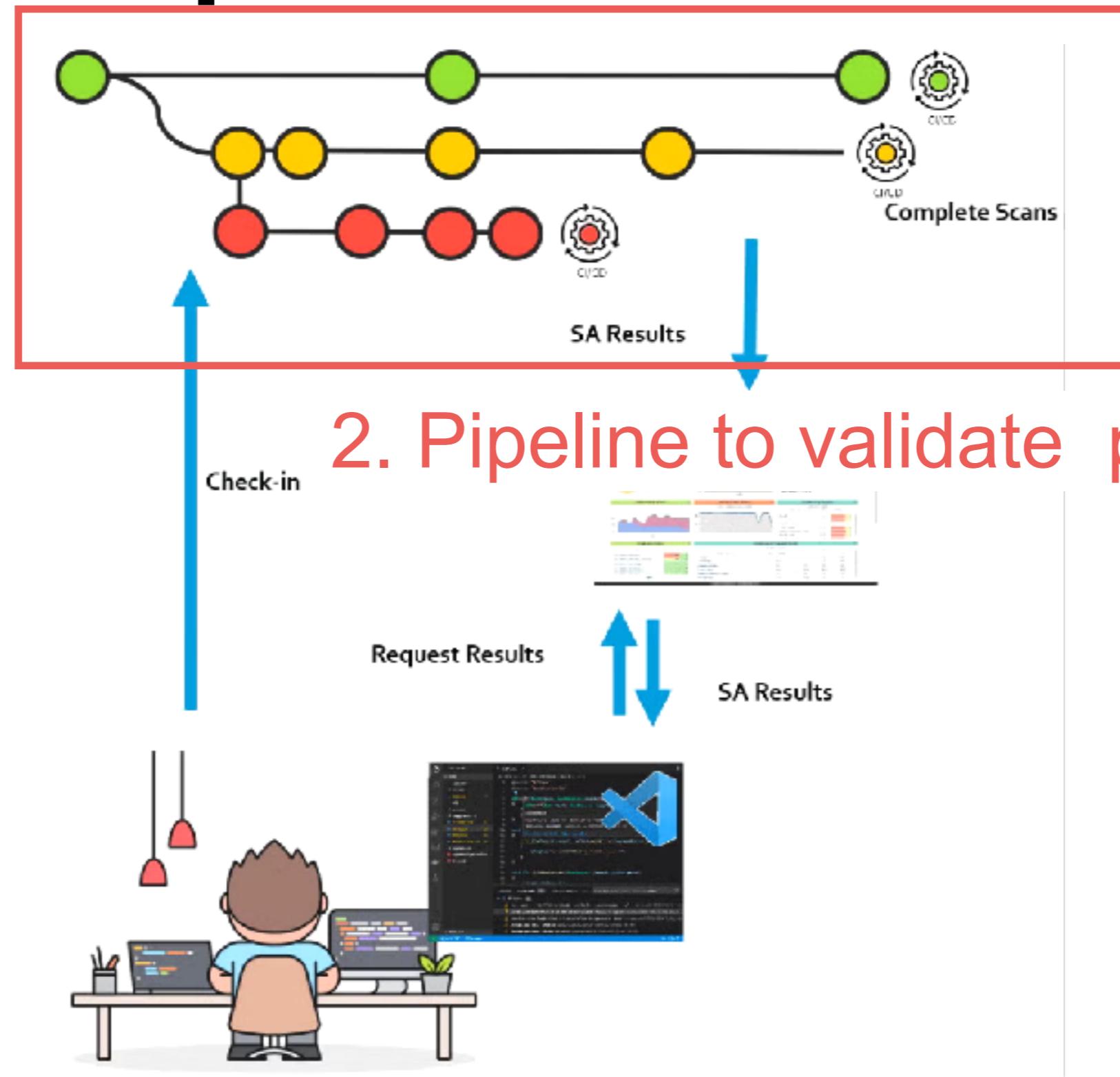
<https://martinfowler.com/articles/continuousIntegration.html>



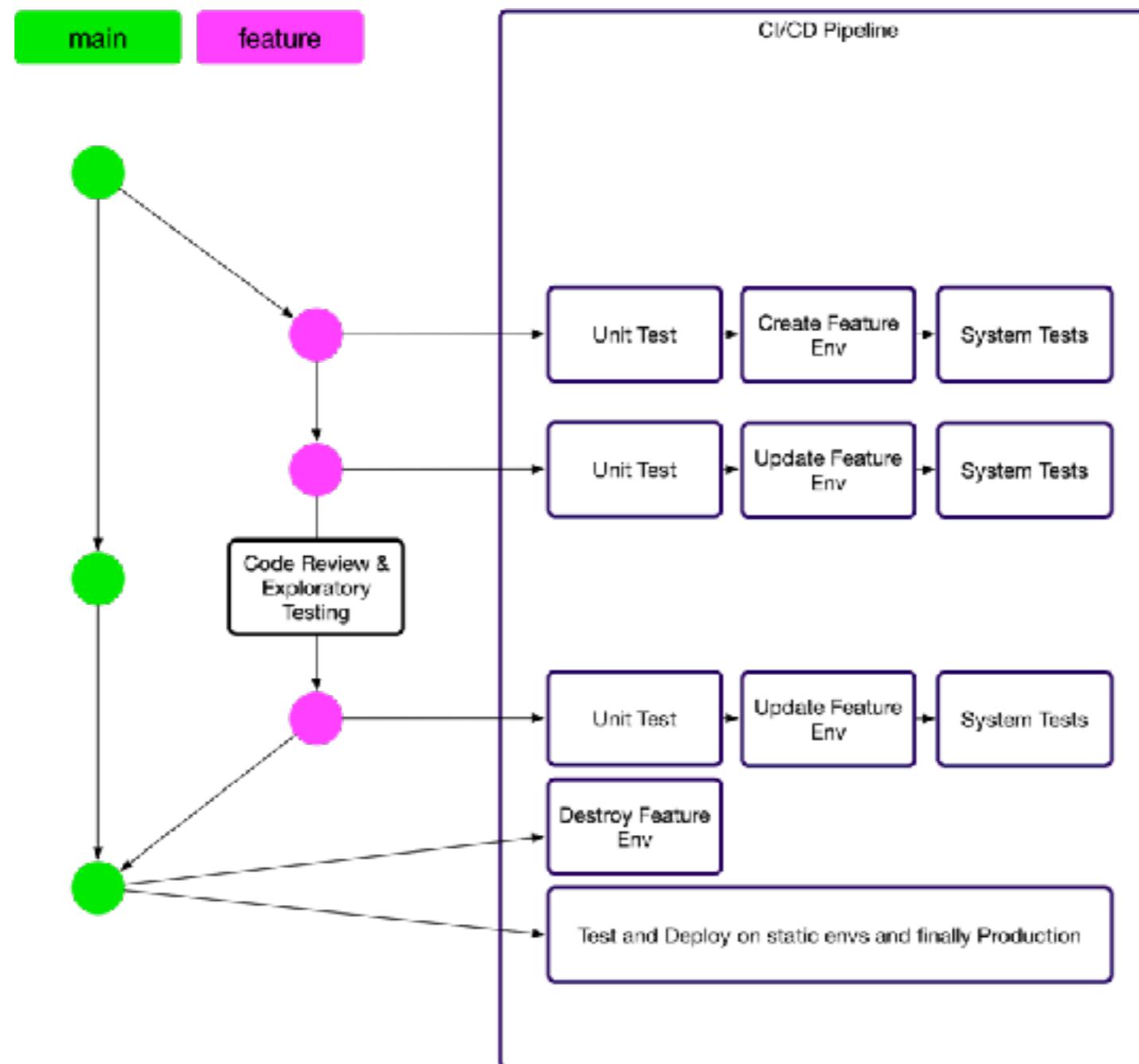
# Development workflow (1)



# Development workflow (3)



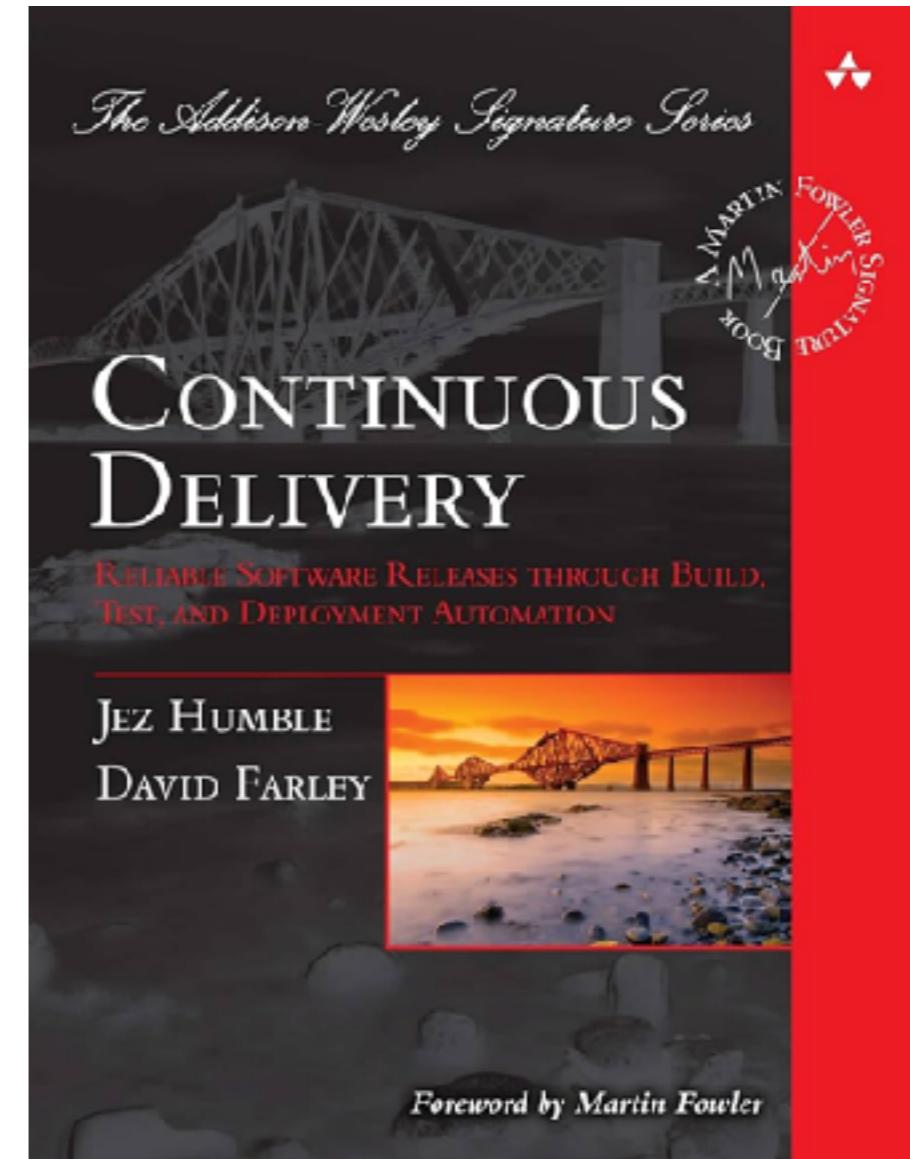
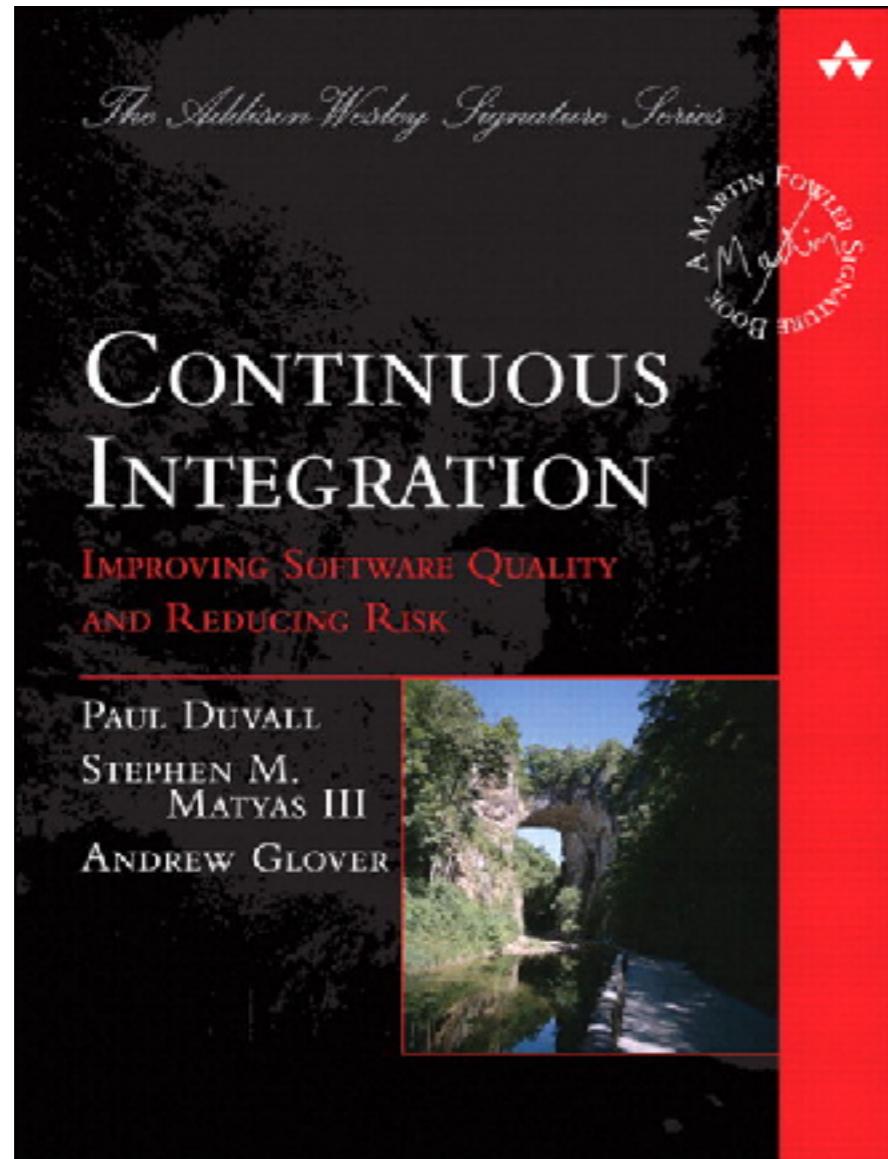
# Development Workflow (1)



# Ready to deliver to someone !!



# Improve quality and Reduce risk



# State of DevOps Report

**High-performing teams deploy more frequently and have much faster lead times.**



**200x more frequent deployments**



**2,555x shorter lead times**

**They make changes with fewer failures, and recover faster from failures.**



**3x lower change failure rate**

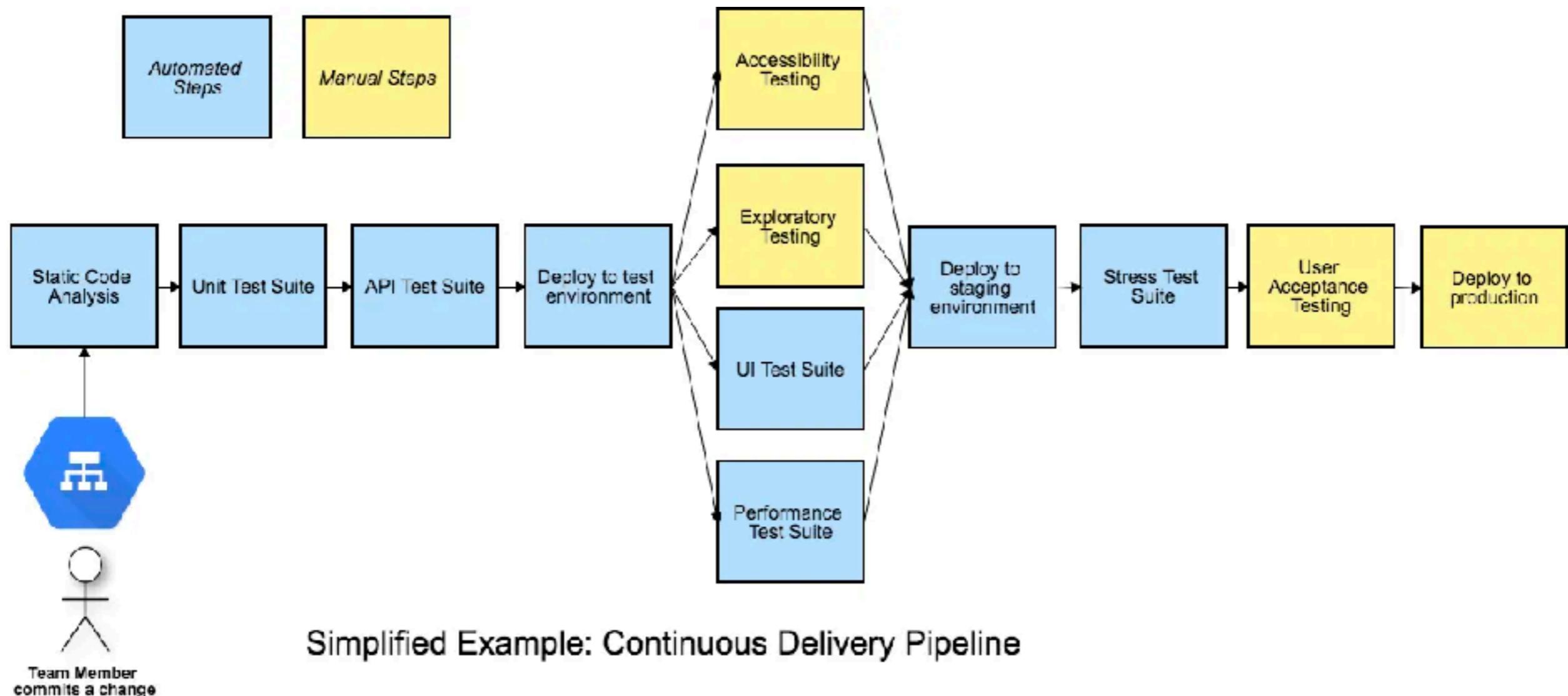


**24x faster recovery from failures**

<https://puppet.com/resources/whitepaper/state-of-devops-report>



# Design your pipeline !!



# GitLab CI

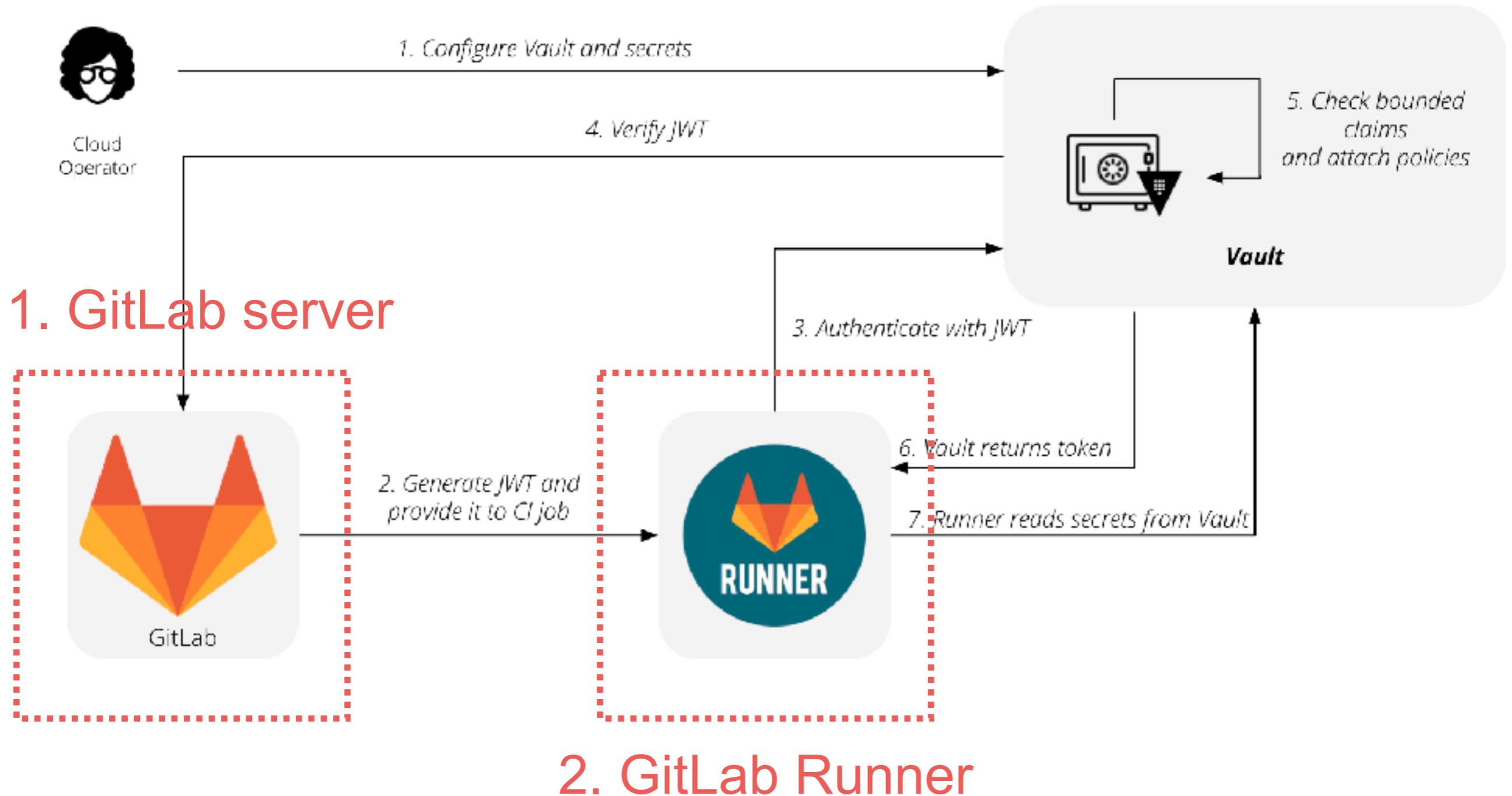
Continuous method of software development,  
where you continuously build, test, deploy, and monitor  
iterative code changes.



<https://docs.gitlab.com/ee/ci/>



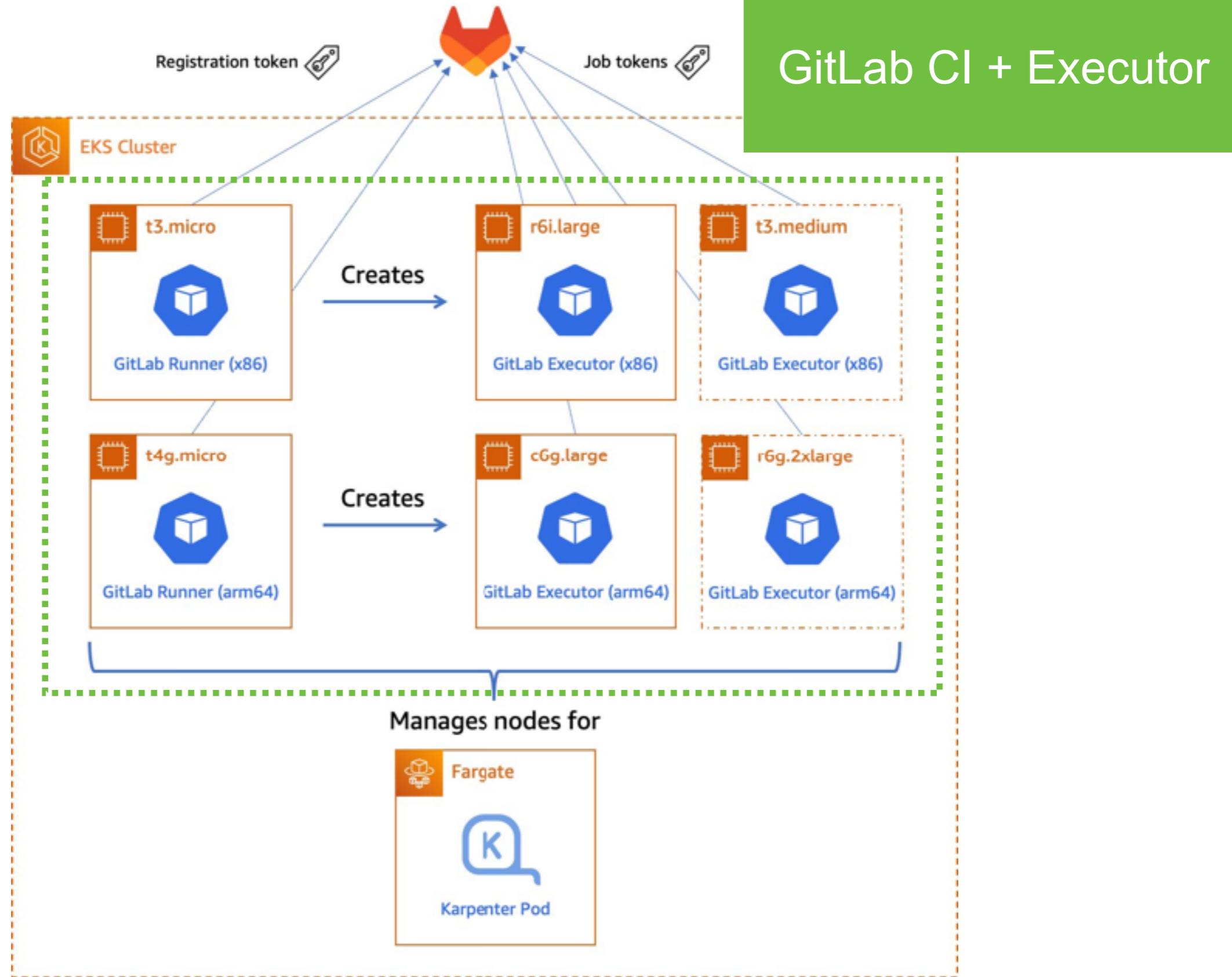
# GitLab CI



Sharing

© 2017 - 2025 Siam Chamnkit Company Limited. All rights reserved.

GitLab API  
(Managed, or self-hosted)



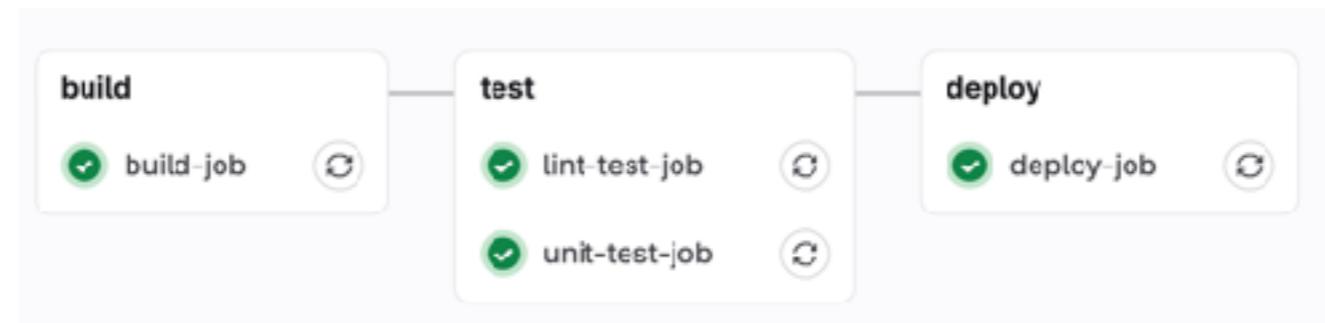
# Create your pipeline

Pipeline-as-code

File **gitlab-ci.yml** in your repository



# Pipeline



```
stages:          # List of stages for jobs, and their order of execution
- build
- test
- deploy

build-job:      # This job runs in the build stage, which runs first.
  stage: build
  script:
    - echo "Compiling the code..."
    - echo "Compile complete."

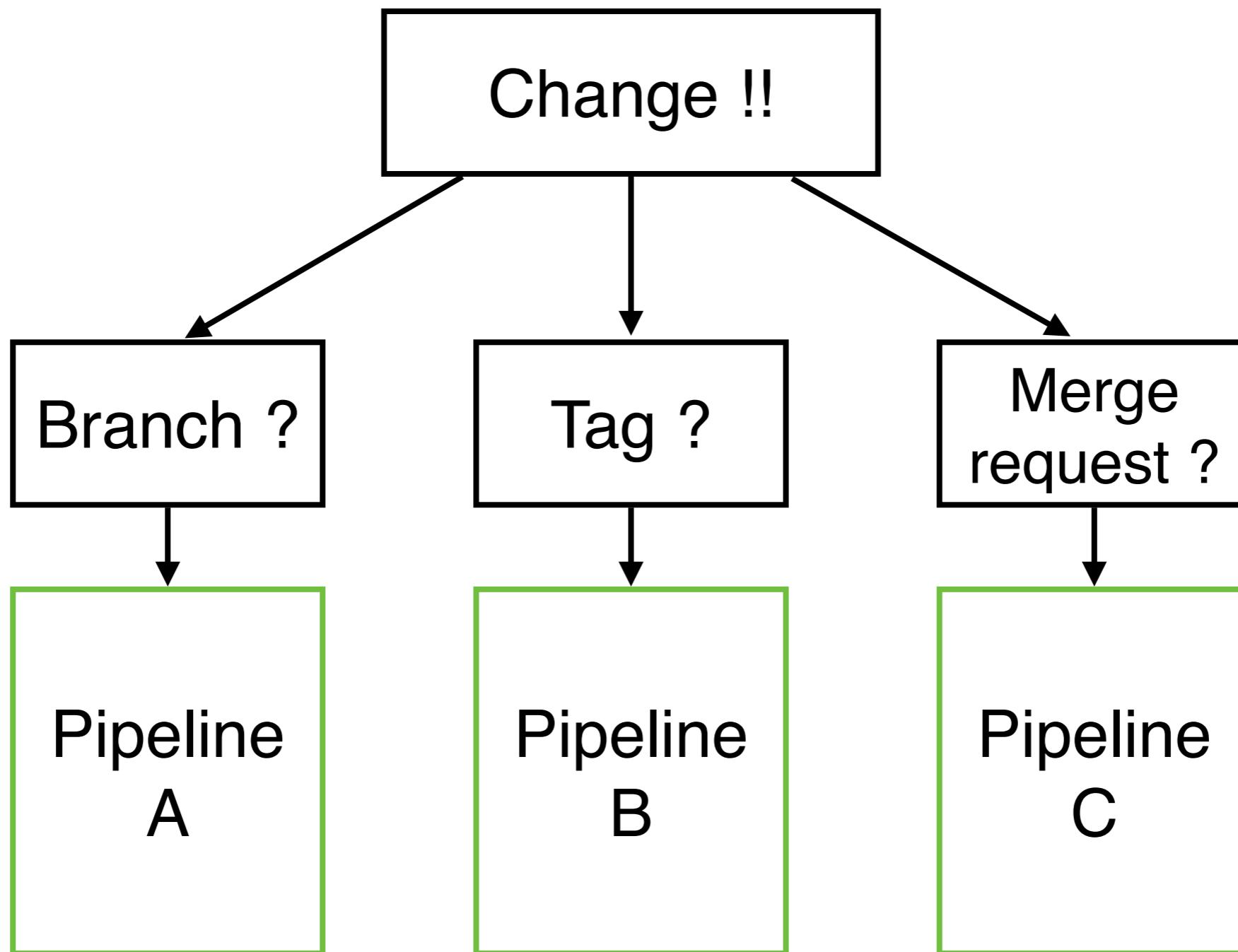
unit-test-job:  # This job runs in the test stage.
  stage: test   # It only starts when the job in the build stage completes successfully.
  script:
    - echo "Running unit tests... This will take about 60 seconds."
    - echo "Code coverage is 98%"

lint-test-job:  # This job also runs in the test stage.
  stage: test   # It can run at the same time as unit-test-job (in parallel).
  script:
    - echo "Linting code... This will take about 10 seconds."
    - echo "No lint issues found."

deploy-job:     # This job runs in the deploy stage.
  stage: deploy # It only runs when *both* jobs in the test stage complete successfully.
  environment: production
  script:
    - echo "Deploying application..."
    - echo "Application successfully deployed."
```



# Create your pipeline



# Condition in pipeline (1)

```
unit-test-job:  
  stage: test  
  
  rules:  
    - if: '$CI_COMMIT_BRANCH == "main"'  
  
  script:  
    - echo "Running unit tests... This will take about 60  
seconds."  
    - echo "Code coverage is 90%"
```

<https://docs.gitlab.com/ee/ci/>



# Condition in pipeline (2)

Merge request + tag + branch

```
workflow:  
  rules:  
    - if: $CI_PIPELINE_SOURCE == 'merge_request_event'  
    - if: $CI_COMMIT_TAG  
    - if: $CI_COMMIT_BRANCH == $CI_DEFAULT_BRANCH
```

[https://docs.gitlab.com/ee/ci/pipelines/merge\\_request\\_pipelines.html](https://docs.gitlab.com/ee/ci/pipelines/merge_request_pipelines.html)



# Demo









# State of DevOps Report

**High-performing teams deploy more frequently and have much faster lead times.**



**200x more frequent deployments**



**2,555x shorter lead times**

**They make changes with fewer failures, and recover faster from failures.**



**3x lower change failure rate**



**24x faster recovery from failures**

<https://puppet.com/resources/whitepaper/state-of-devops-report>



# **What is DevOps ?**



# DevOps is not ...

Certification

Role

Set of tools

Prescriptive process



# **Gitlab is CI/CD ?**



# **Continuous Integration Continuous Delivery**



# Why CI/CD ?

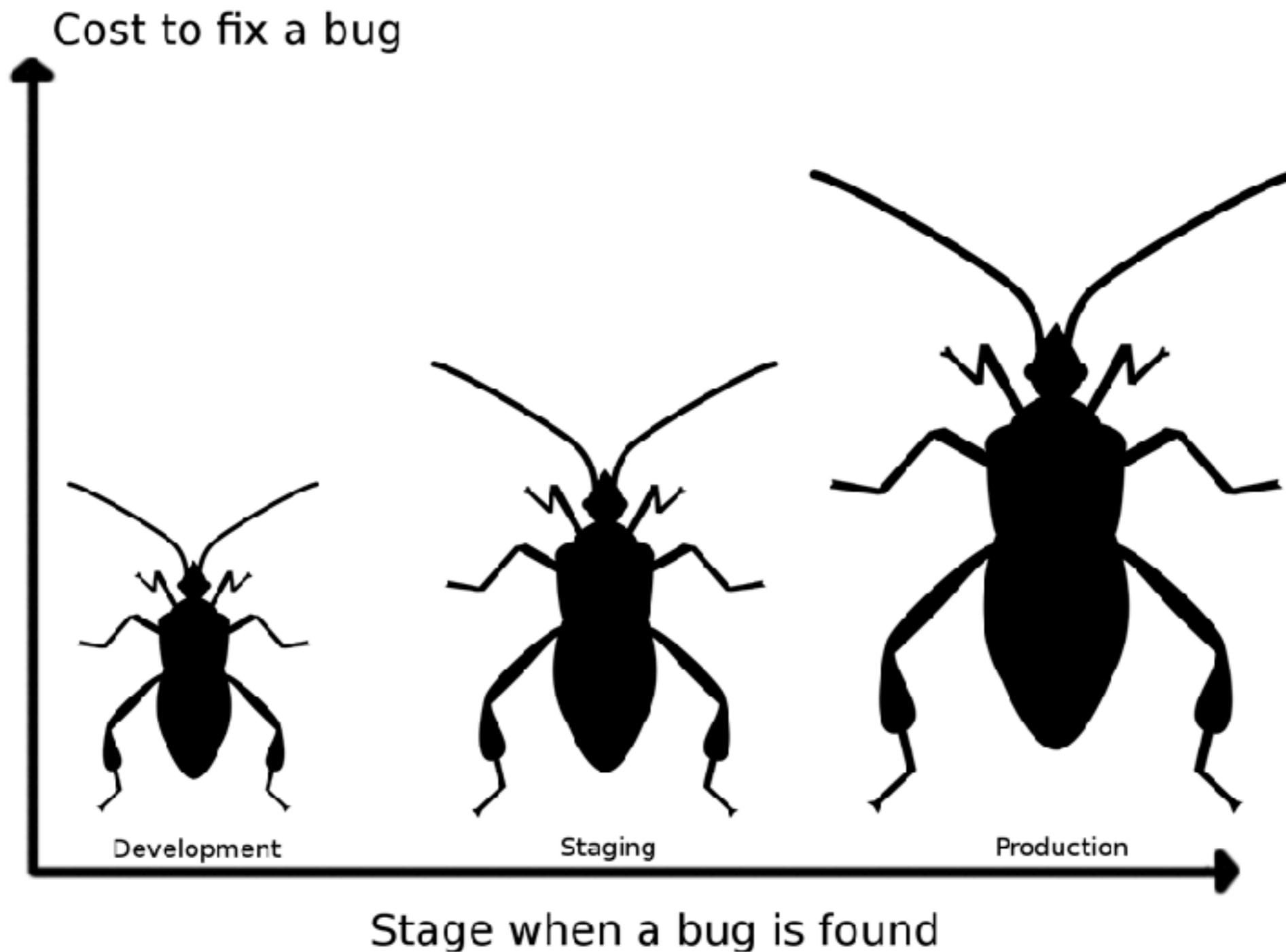


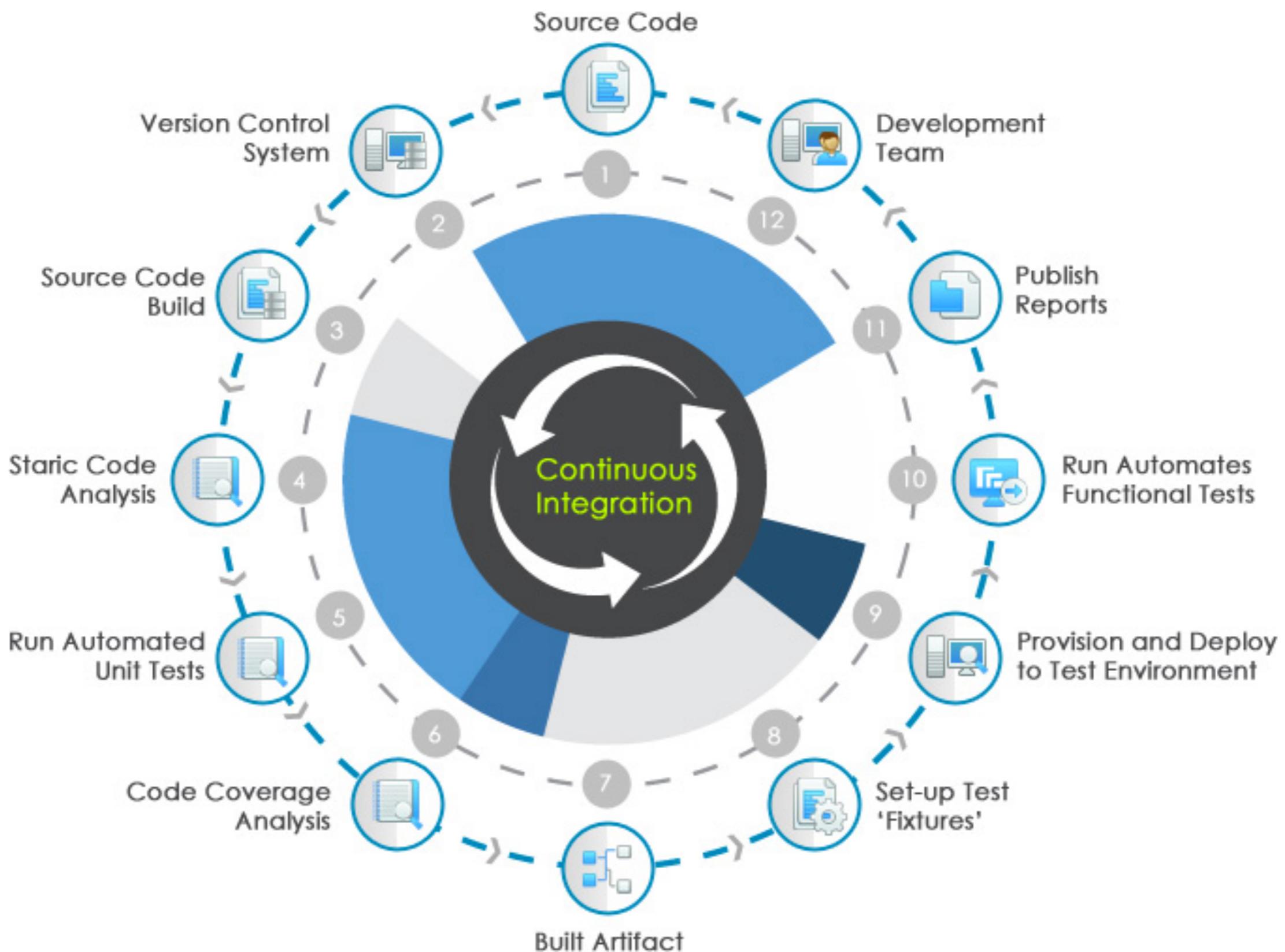
# The cost of integration

1. Merging the code
2. Duplicate changes
3. Test again again !!
4. Fixing bugs
5. Impact on stability



# The cost of integration







Jenkins

Bamboo



TeamCity

> go<sup>TM</sup>



Hudson





Jenkins

Bamboo

CI is about what people do  
not about what tools they use



Visual Studio



Team Foundation Server

Hudson



Travis

wercker

circleci



# Continuous Integration

Discipline to integrate frequently



# Continuous Integration

Strive to make **small change**



# Continuous Integration

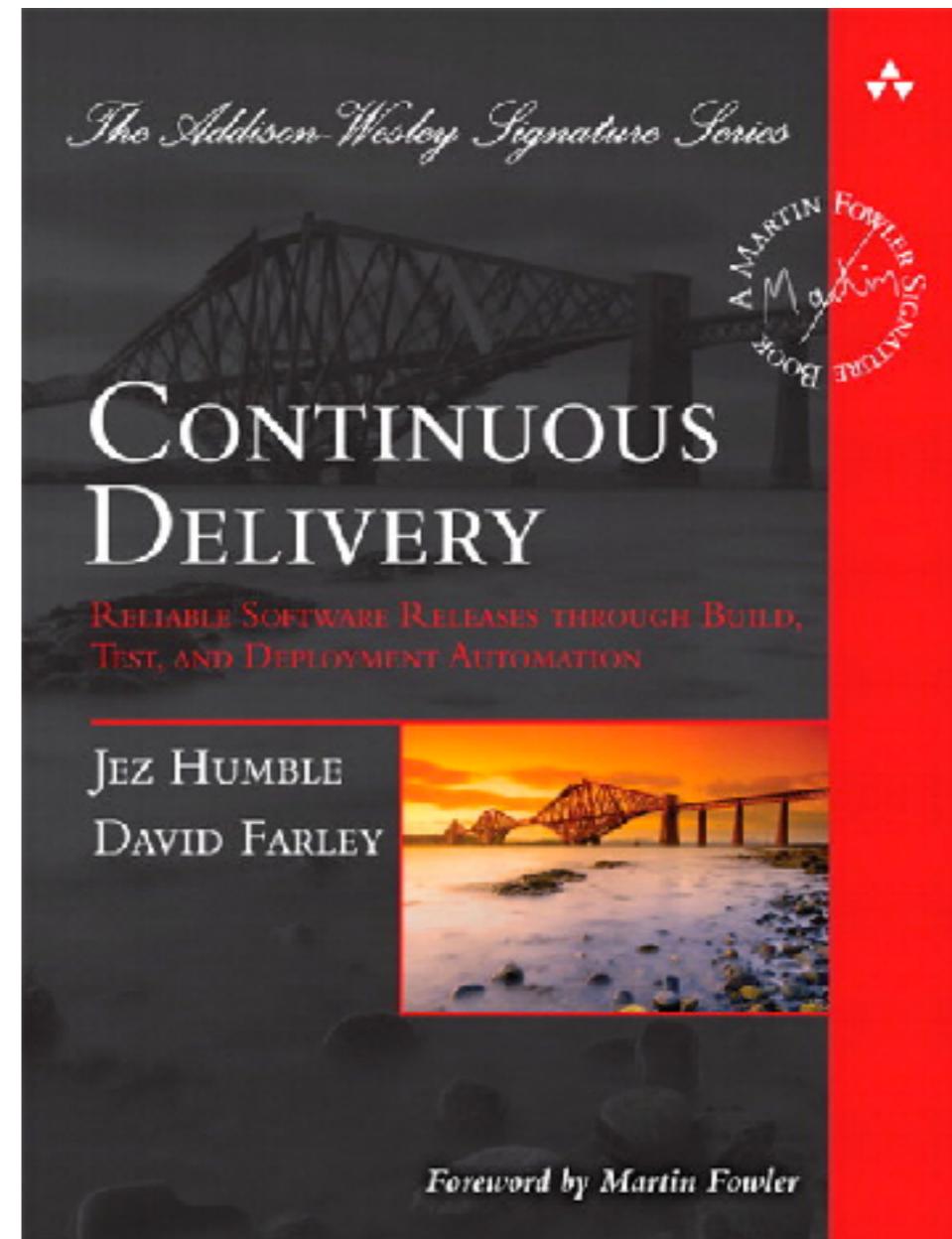
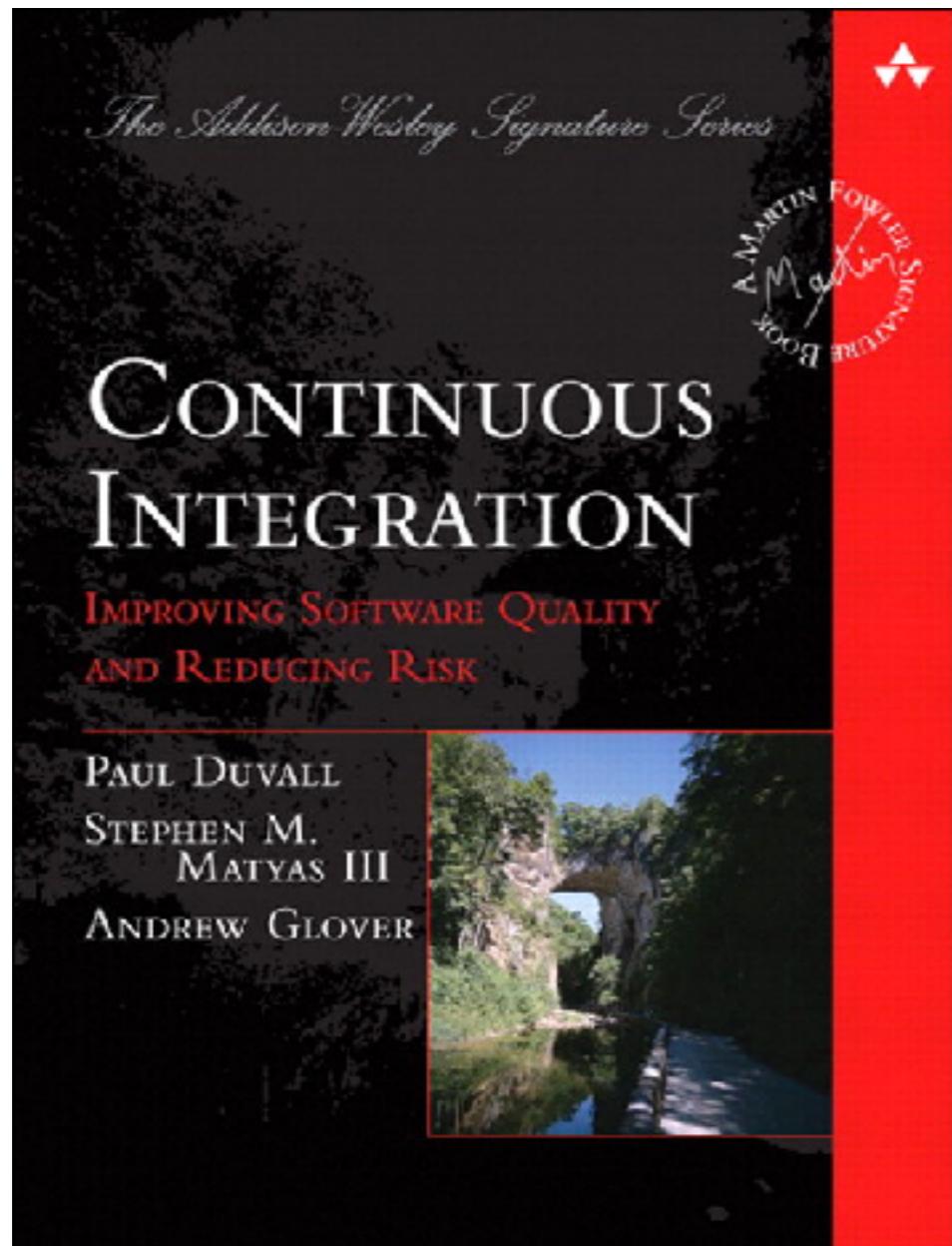
Strive for **fast feedback**



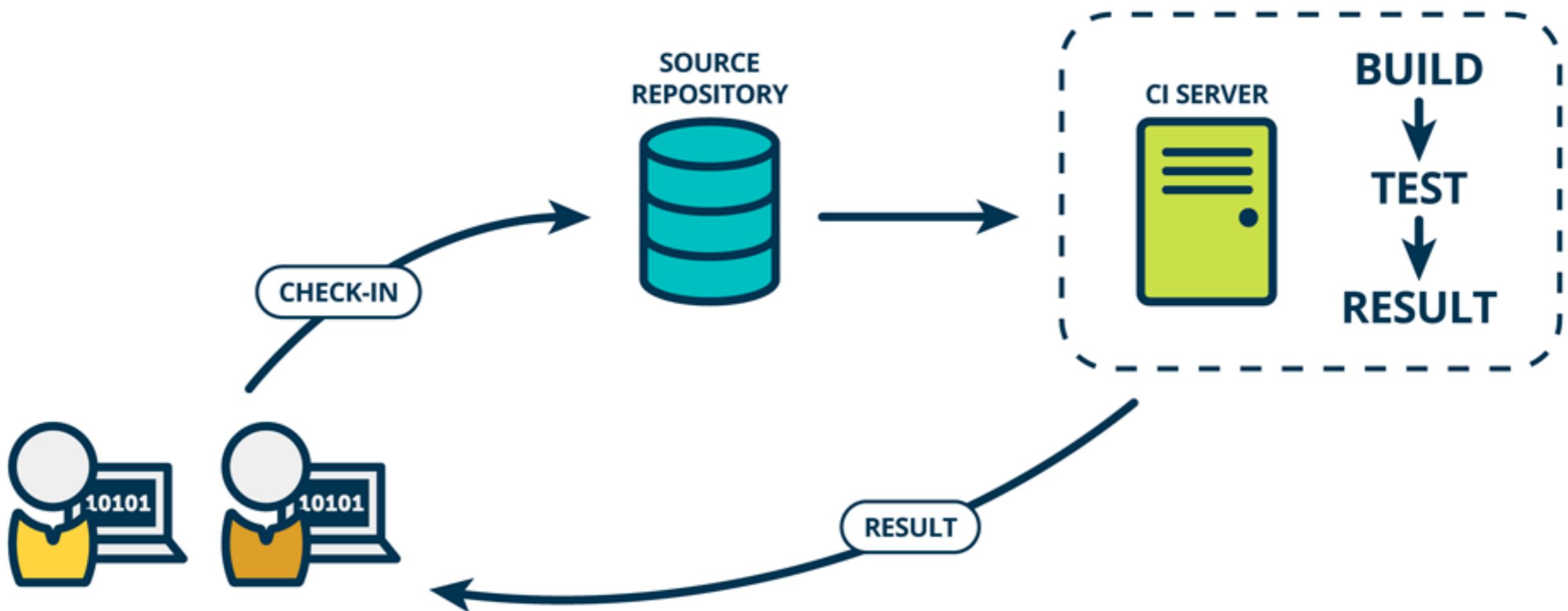
# **Practices of Continuous Integration**



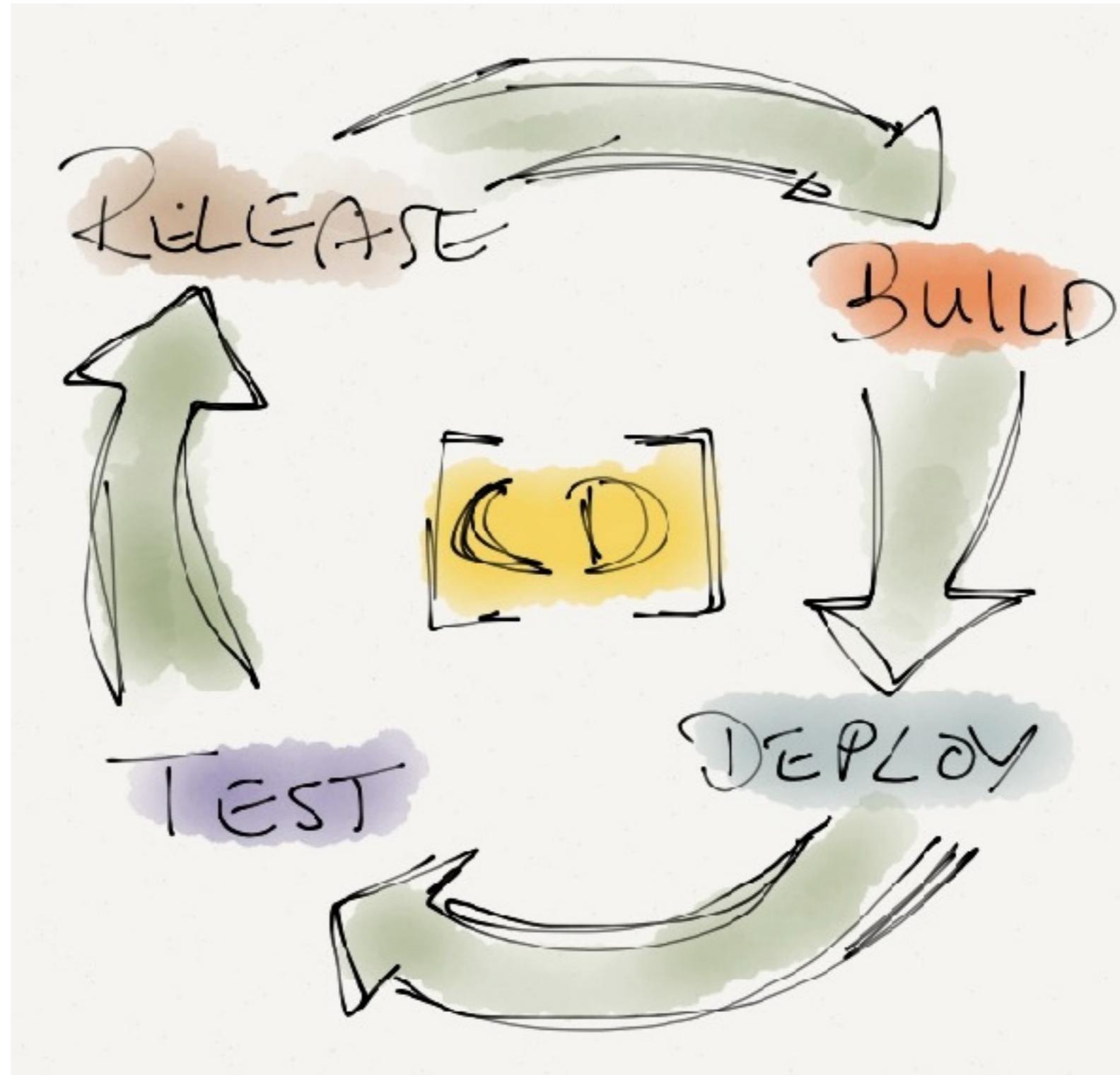
# Improve quality and reduce risk



# Continuous Integration



# CD ?



# CD ?

## CONTINUOUS DELIVERY



## CONTINUOUS DEPLOYMENT



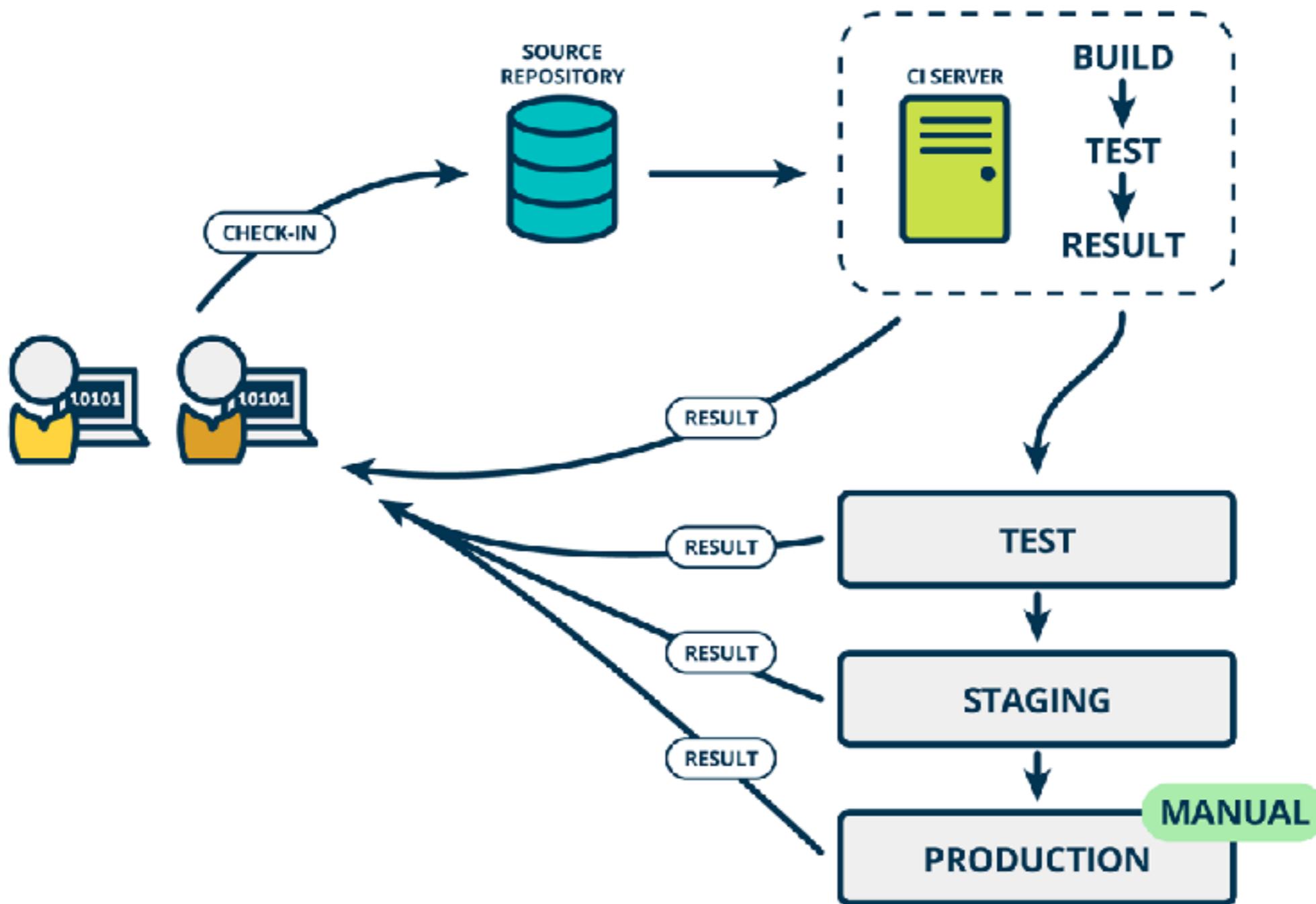
<http://blog.crisp.se/2013/02/05/yassalsundman/continuous-delivery-vs-continuous-deployment>



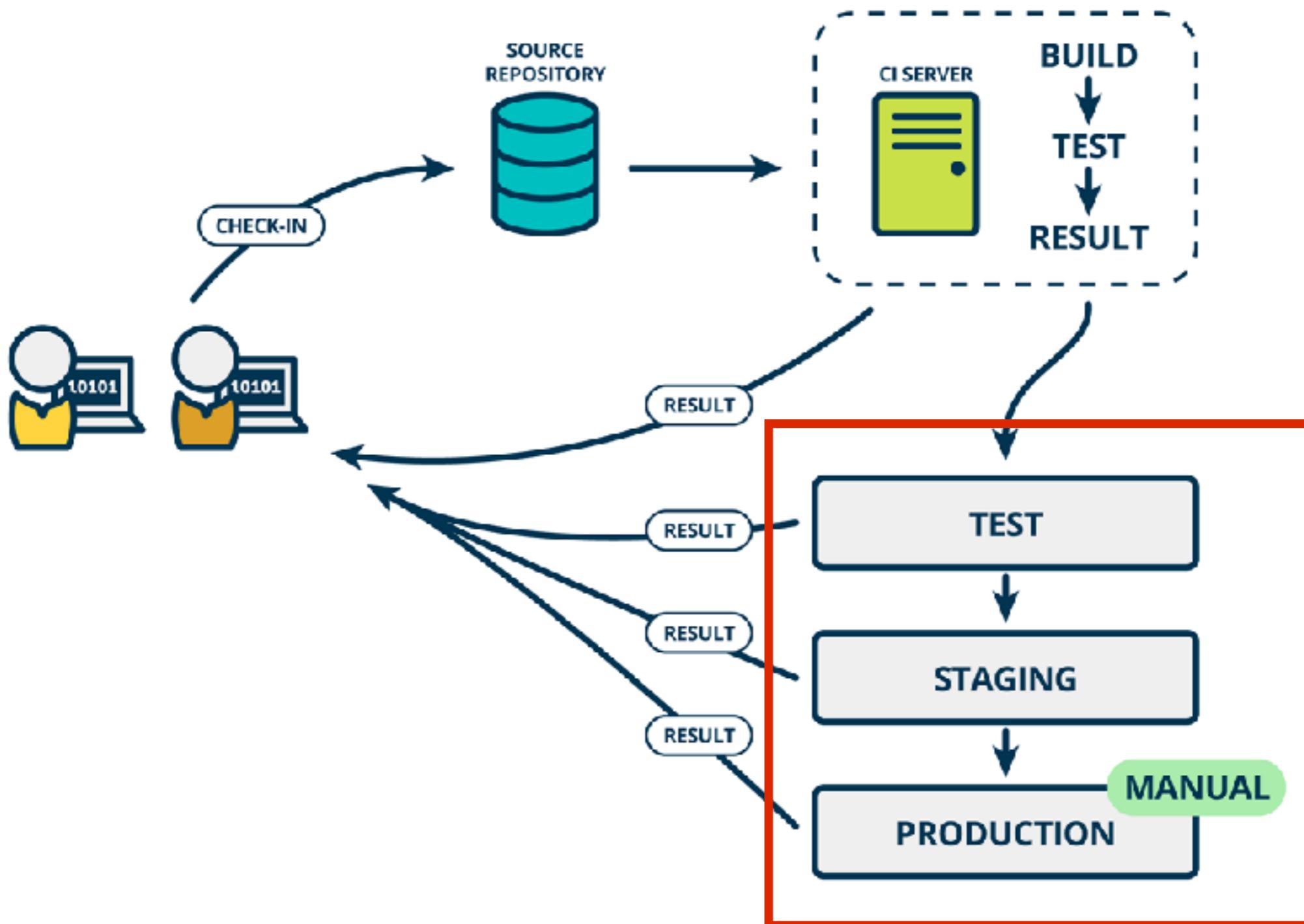
Sharing

© 2017 - 2025 Siam Chamnankit Company Limited. All rights reserved.

# Continuous Delivery



# Rise of DevOps



# **Continuous Integration**

**is a Software development practices**



# Practice 1

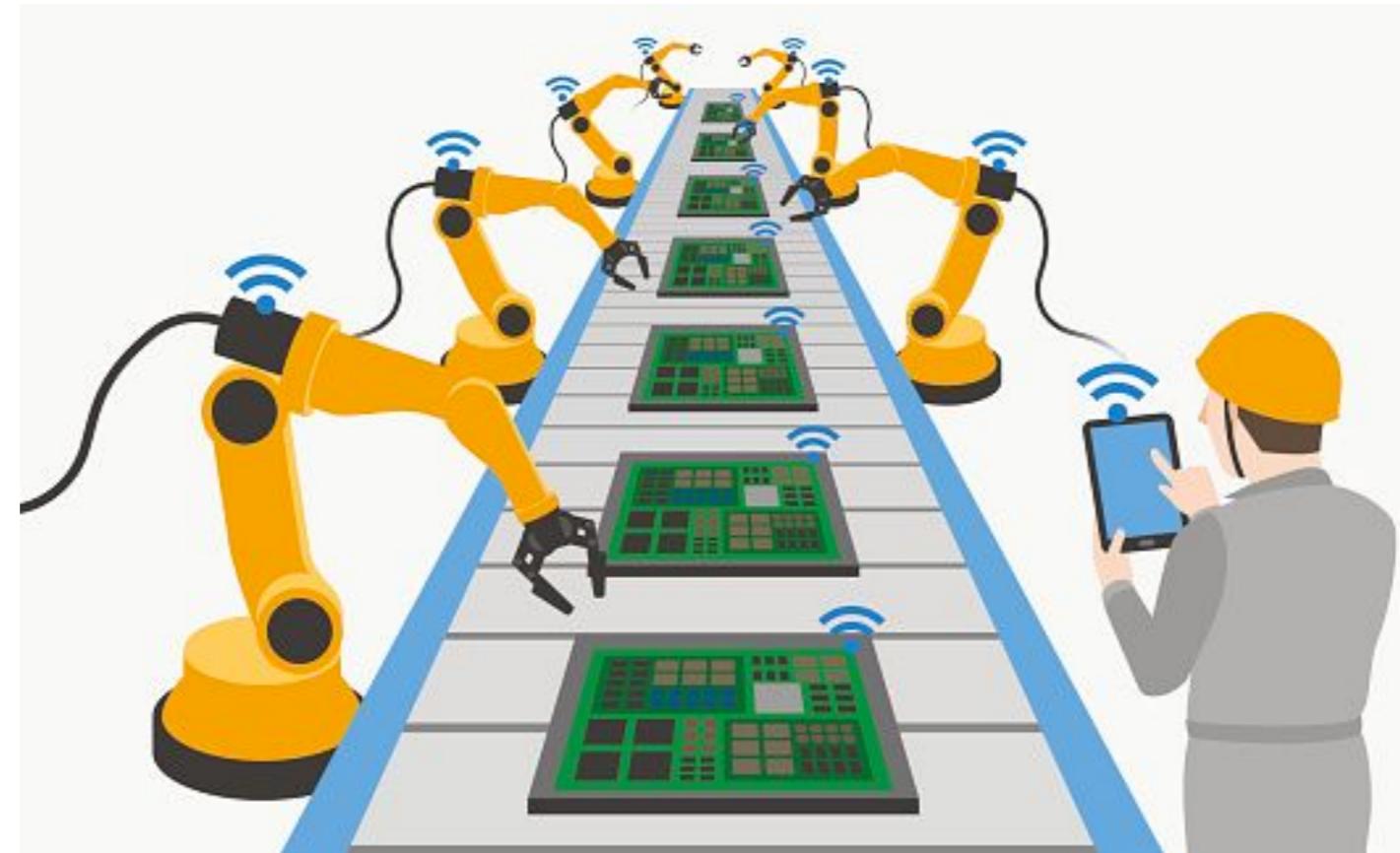
Maintain a single source repository

In general, you should store in source control  
everything you need to build anything



# Practice 2

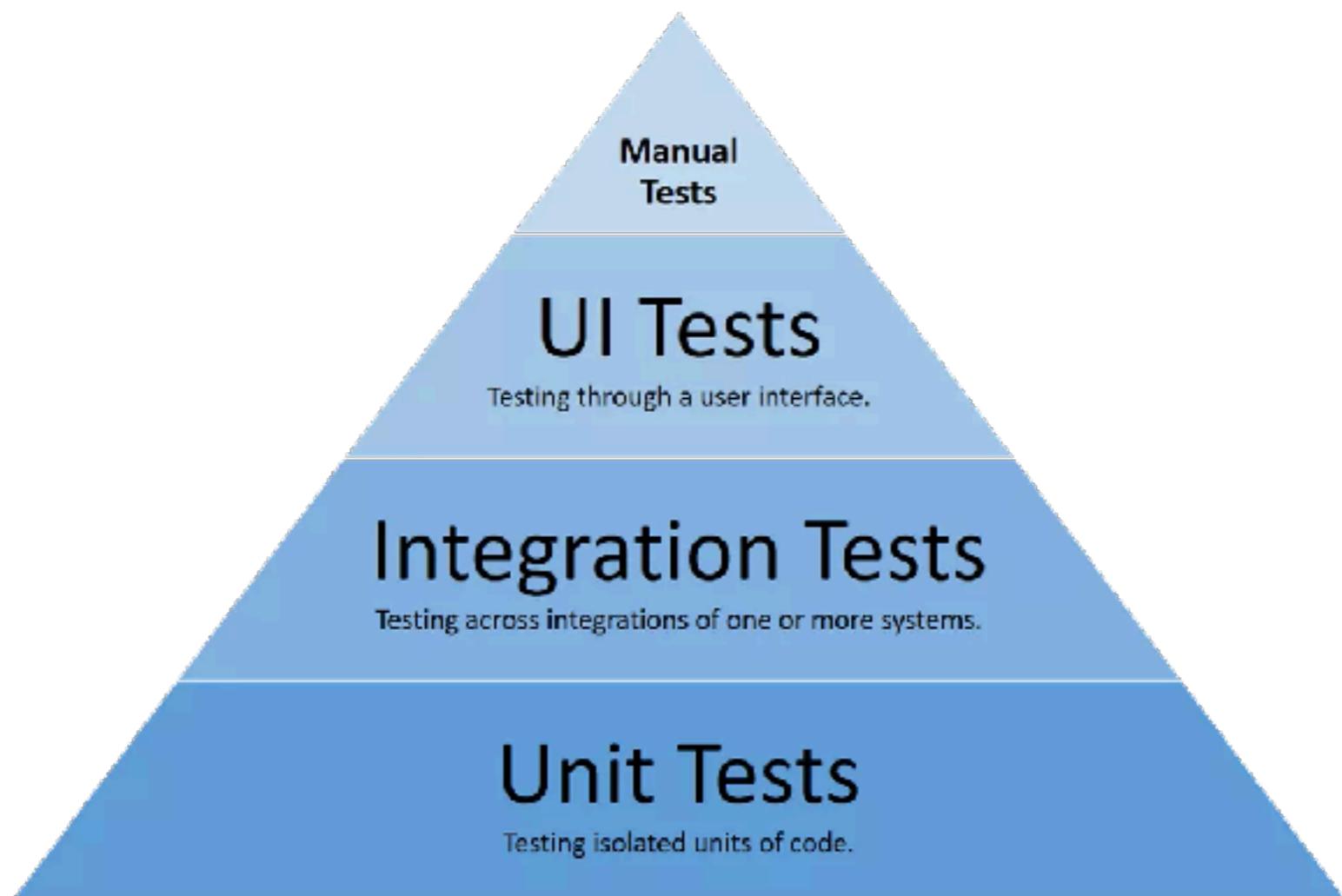
Automated the build  
Automated environment for builds



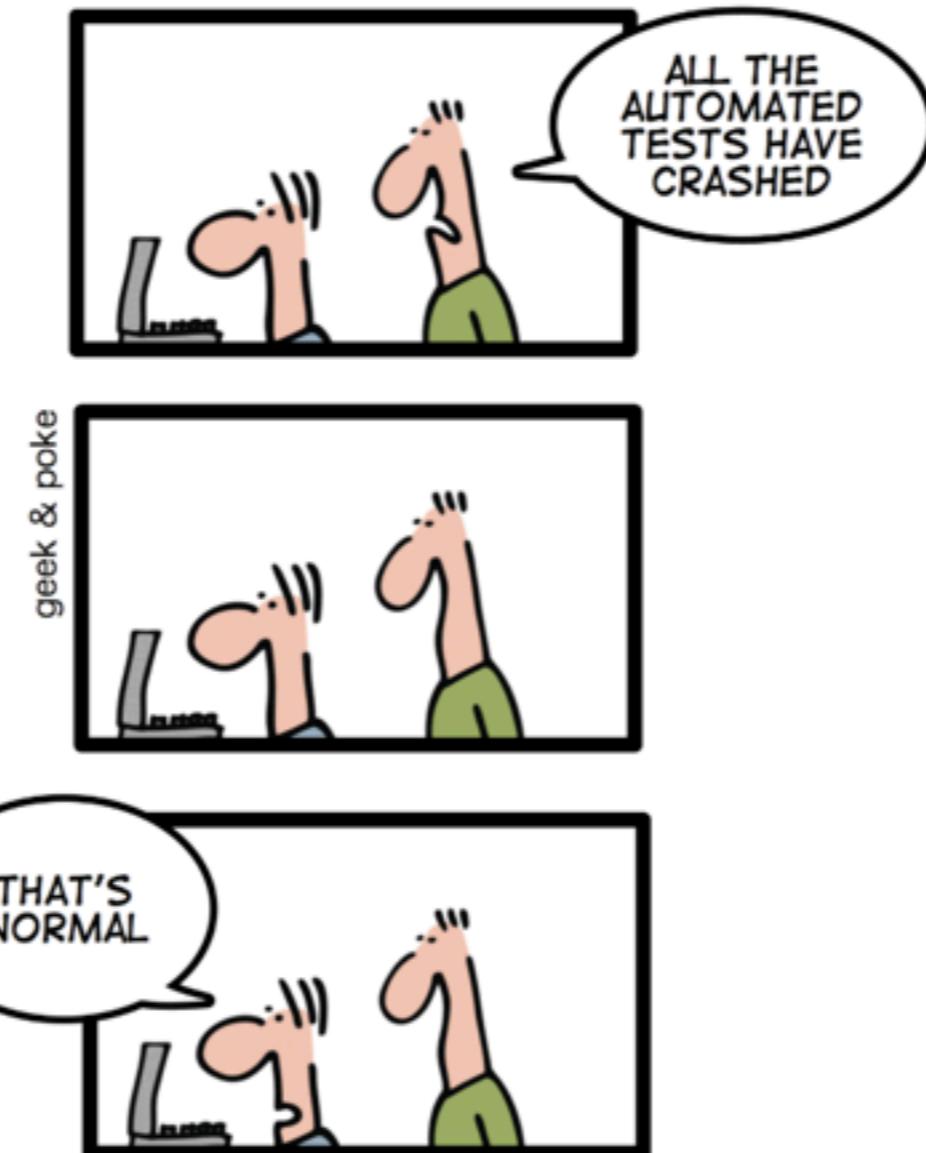
# Practice 3

Make your build **self-testing**

Build process => compile, linking and **testing**



*TODAY: CONTINUOUS INTEGRATION  
GIVES YOU THE COMFORTING  
FEELING TO KNOW THAT  
EVERYTHING IS NORMAL*



<http://geekandpoke.typepad.com/>

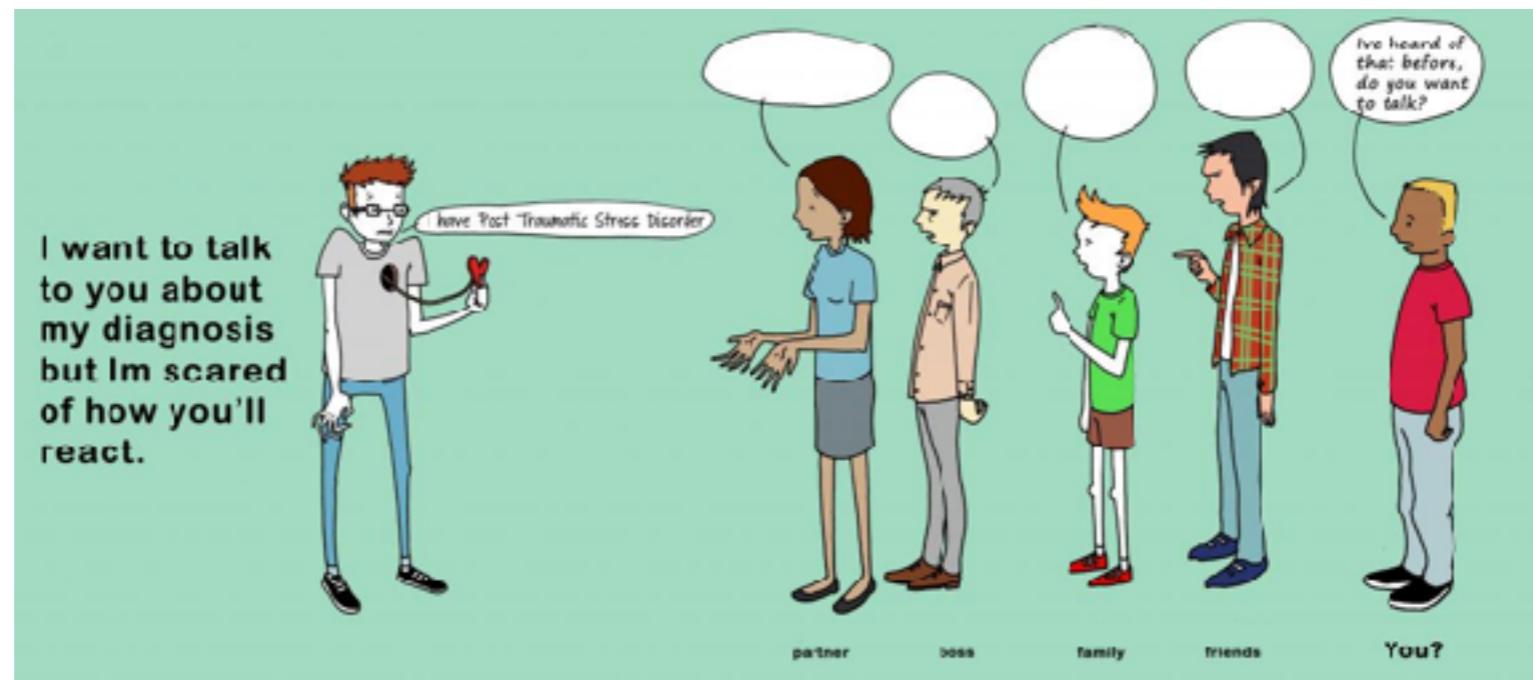


# Practice 4

**Everyone commits to the mainline everyday**

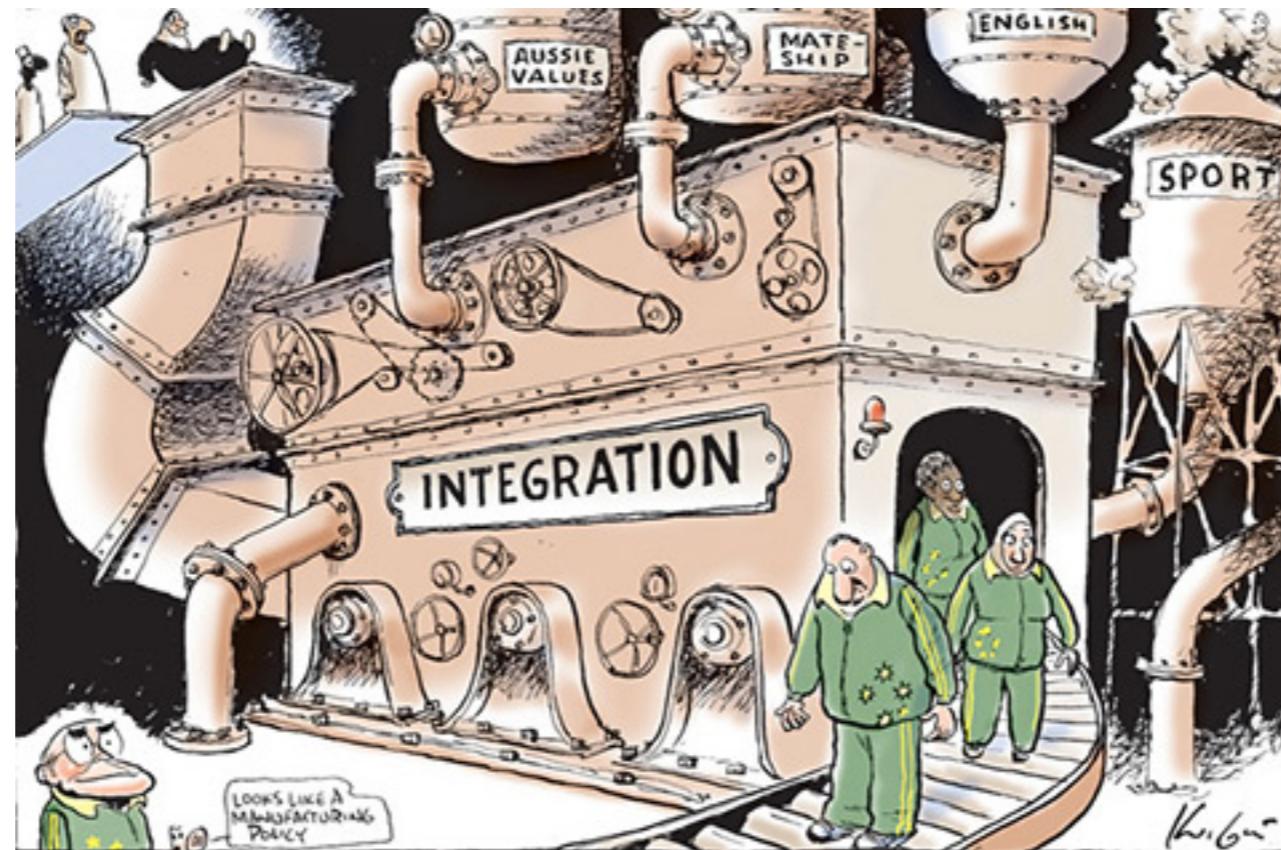
**Integration is about communication**

**Integration allows developers to tell other developers**



# Practice 5

Every commits should build the mainline on an  
**Integration machine**



# Nightly build is not enough for Continuous Integration



# Practice 6

**Fix broken builds immediately**

**“Nobody has a higher priority task than  
fixing the build”**



# Practice 7

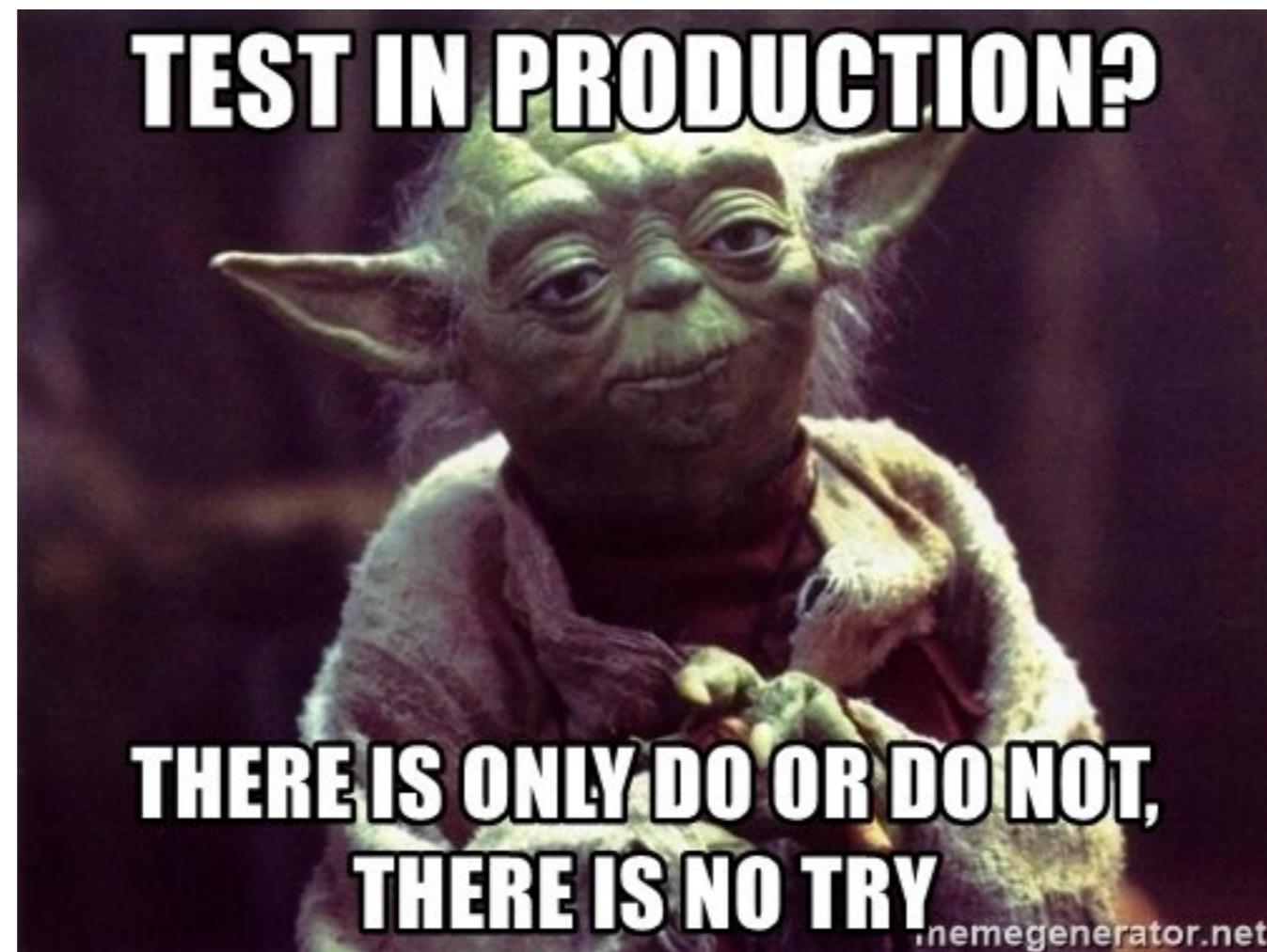
Keep the build **fast**

Continuous Integration is to provide rapid feedback



# Practice 8

Test in clone of the **Production** environment



# Practice 9

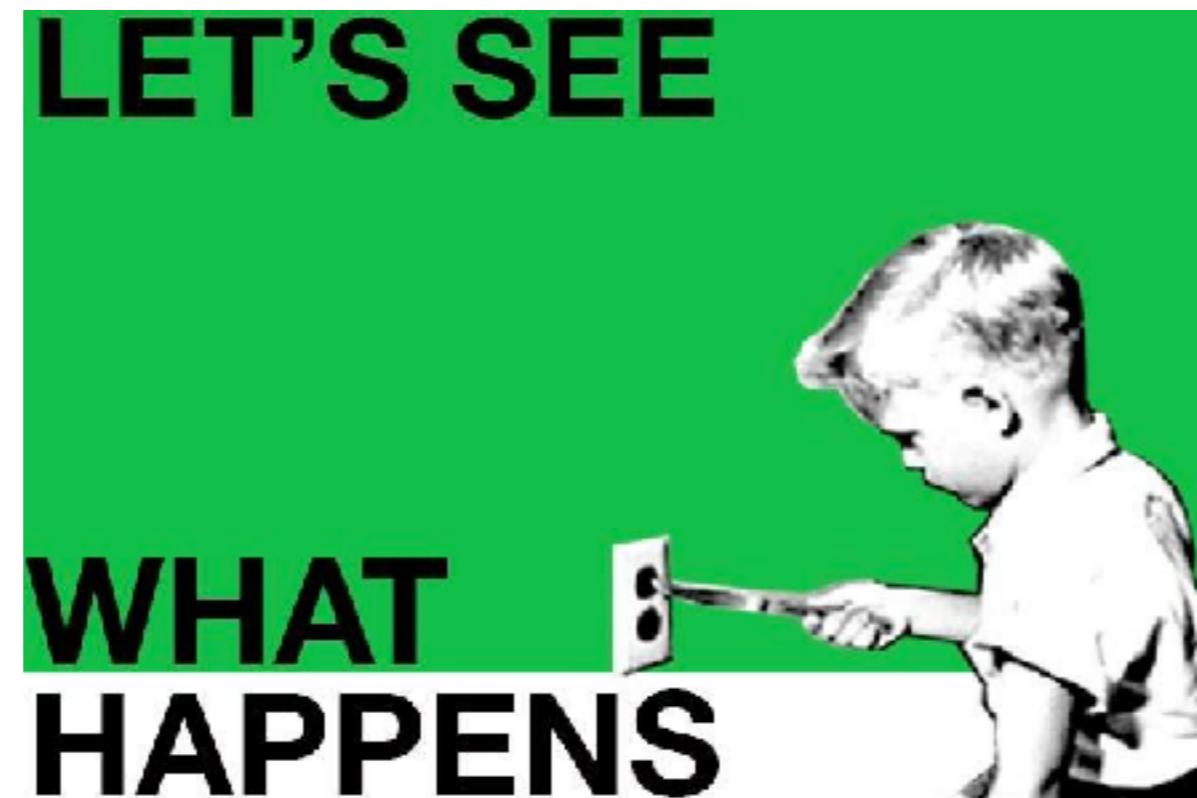
Make it easy for anyone to get  
the latest executable

Make sure well known place where people can find



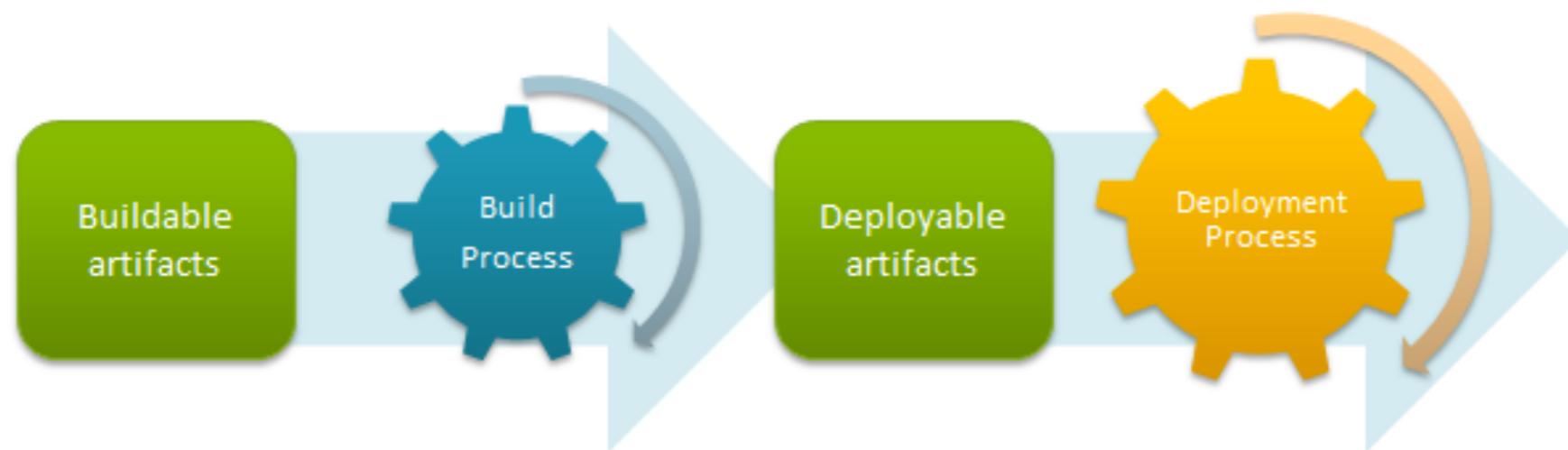
# Practice 10

**Everyone** can see what's happening  
**Easier** to see the state of the system and changes  
Show the good information



# Practice 11

## Automated deployment



# Practice 11

## Automated deployment



# Continuous Delivery



# Continuous Delivery

Use version control for all production artifacts

Automate your deployment process

Implement continuous integration (CI)

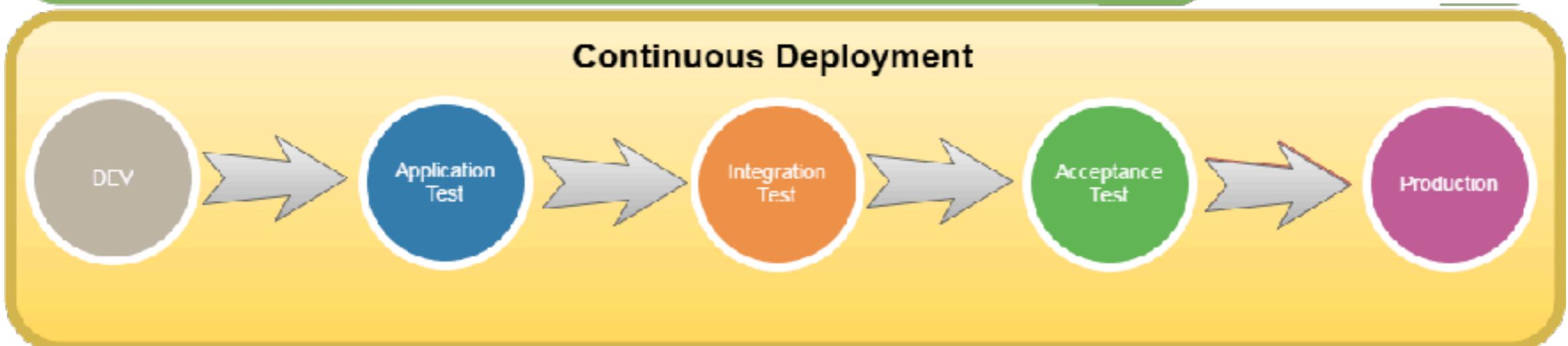
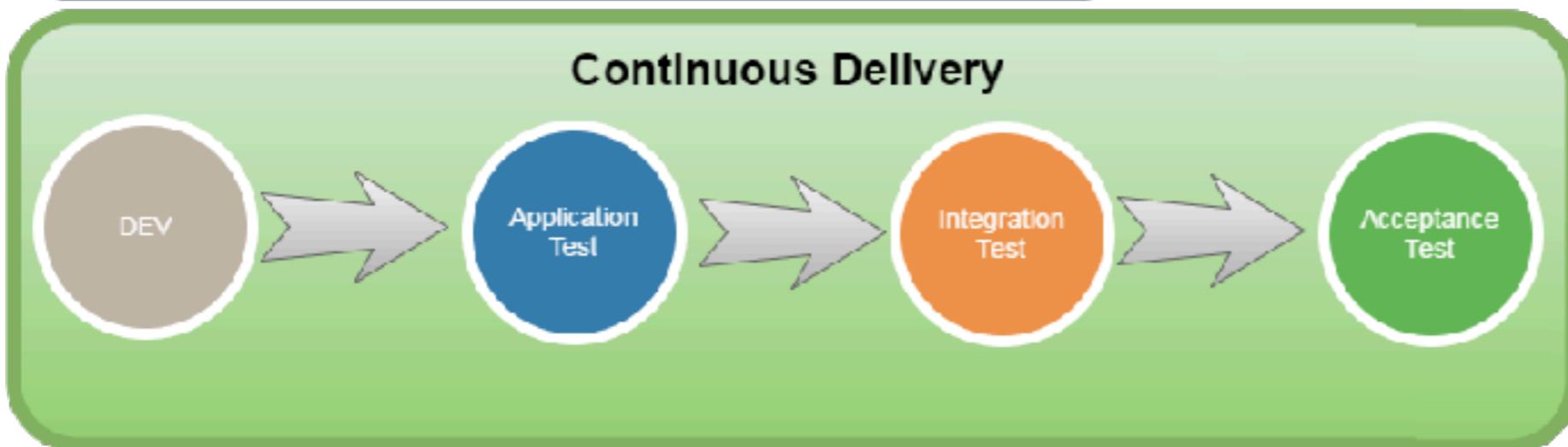
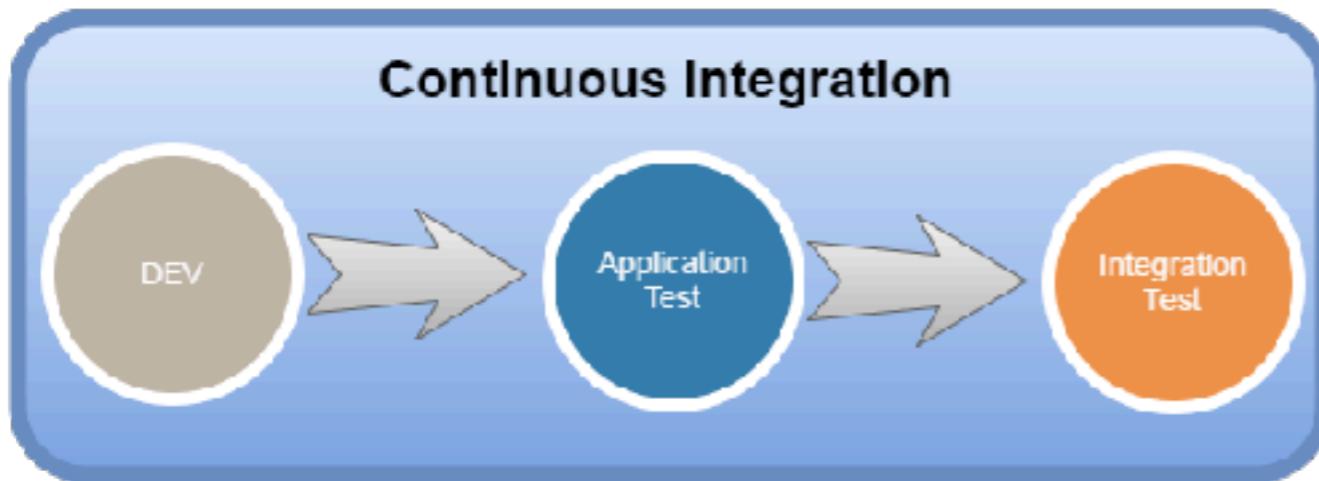
Use trunk-based development methods

Implement test automation

Support test data management

Integrate security into software development process





# **How to achieve the CI ?**



# 1. Use good version control

Local  
Centralize  
Distributed



VSS = A brown粪便 emoji with three wavy lines above it representing steam or smoke.

JUST SAY NO!



## 2. Choose Branch strategy

Main only

Development isolation

Feature isolation

Release isolation

Service and Release isolation

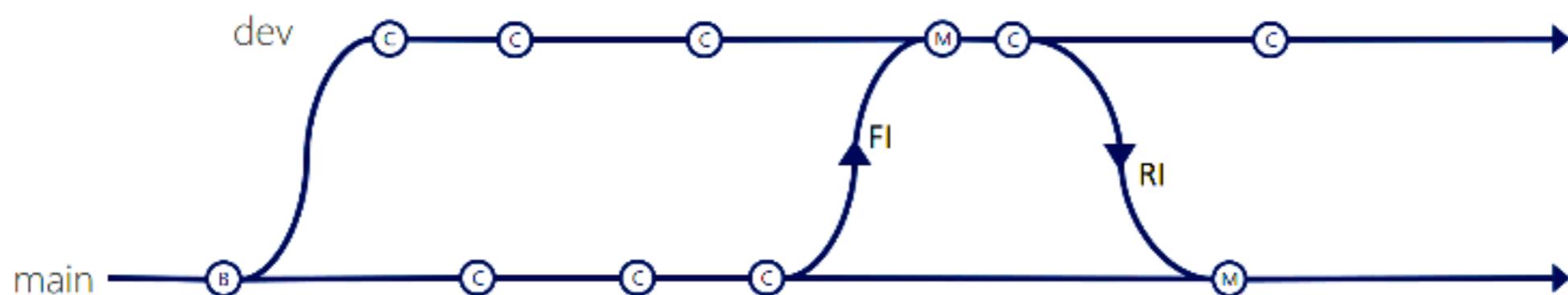
Service, Hotfix and Release isolation



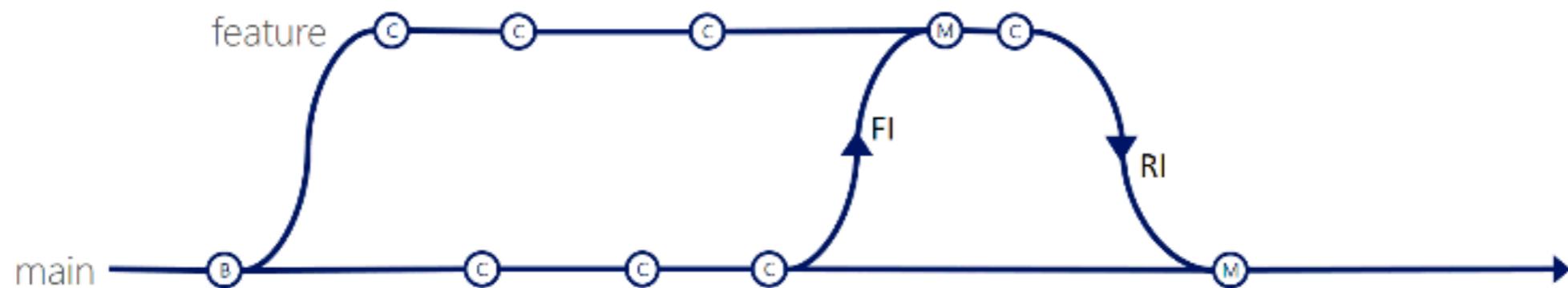
# Main only



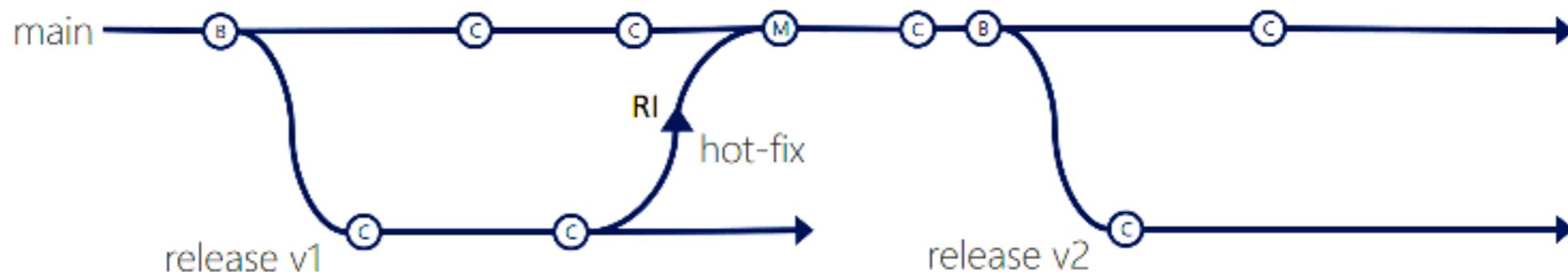
# Development isolation



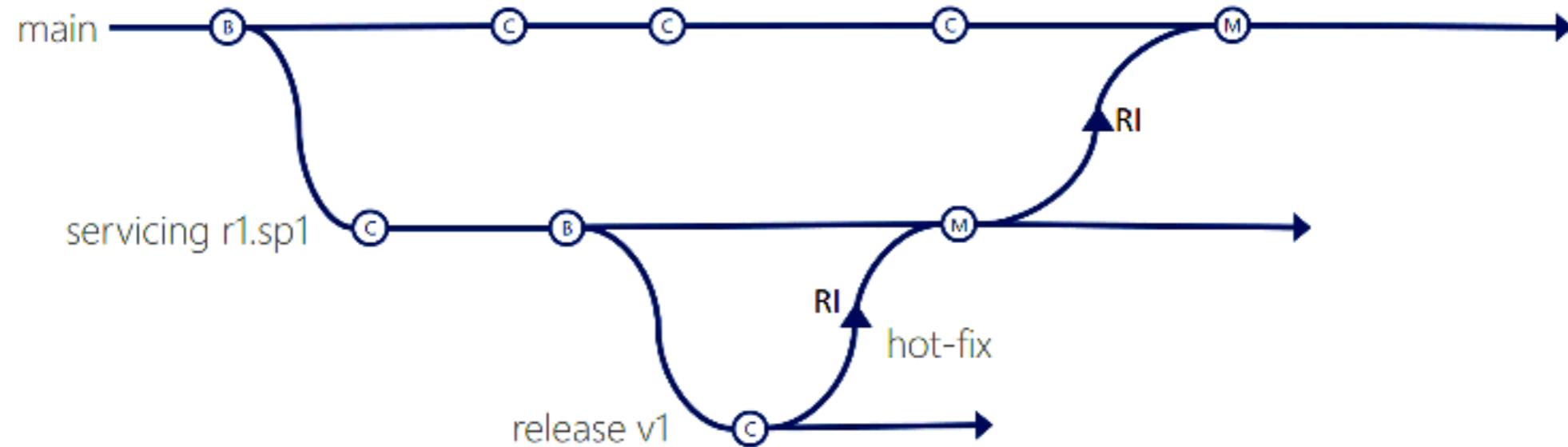
# Feature isolation



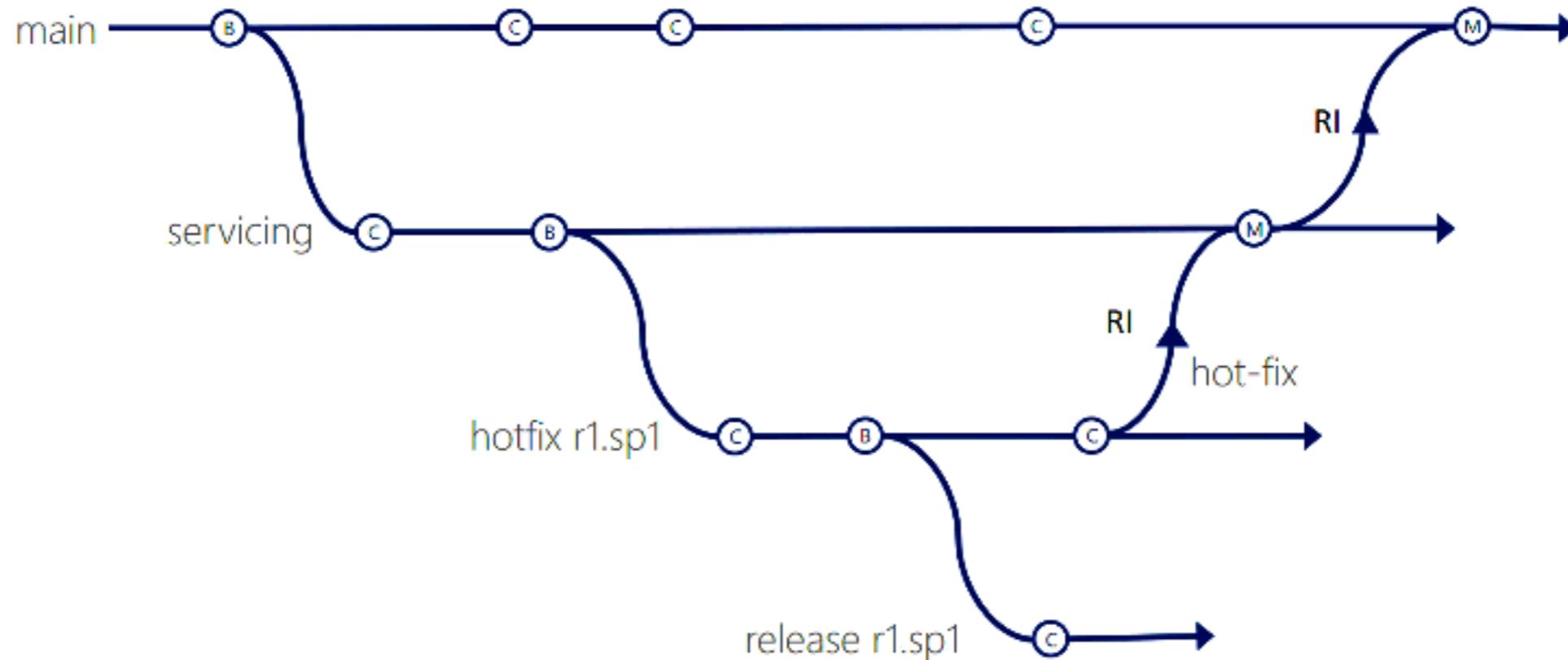
# Release isolation



# Service and Release isolation



# Service, Hotfix, Release isolation



# Workshop

