

# Supercharging Your GitHub Workflow

---

THE MODERN SOFTWARE DEVELOPMENT PIPELINE

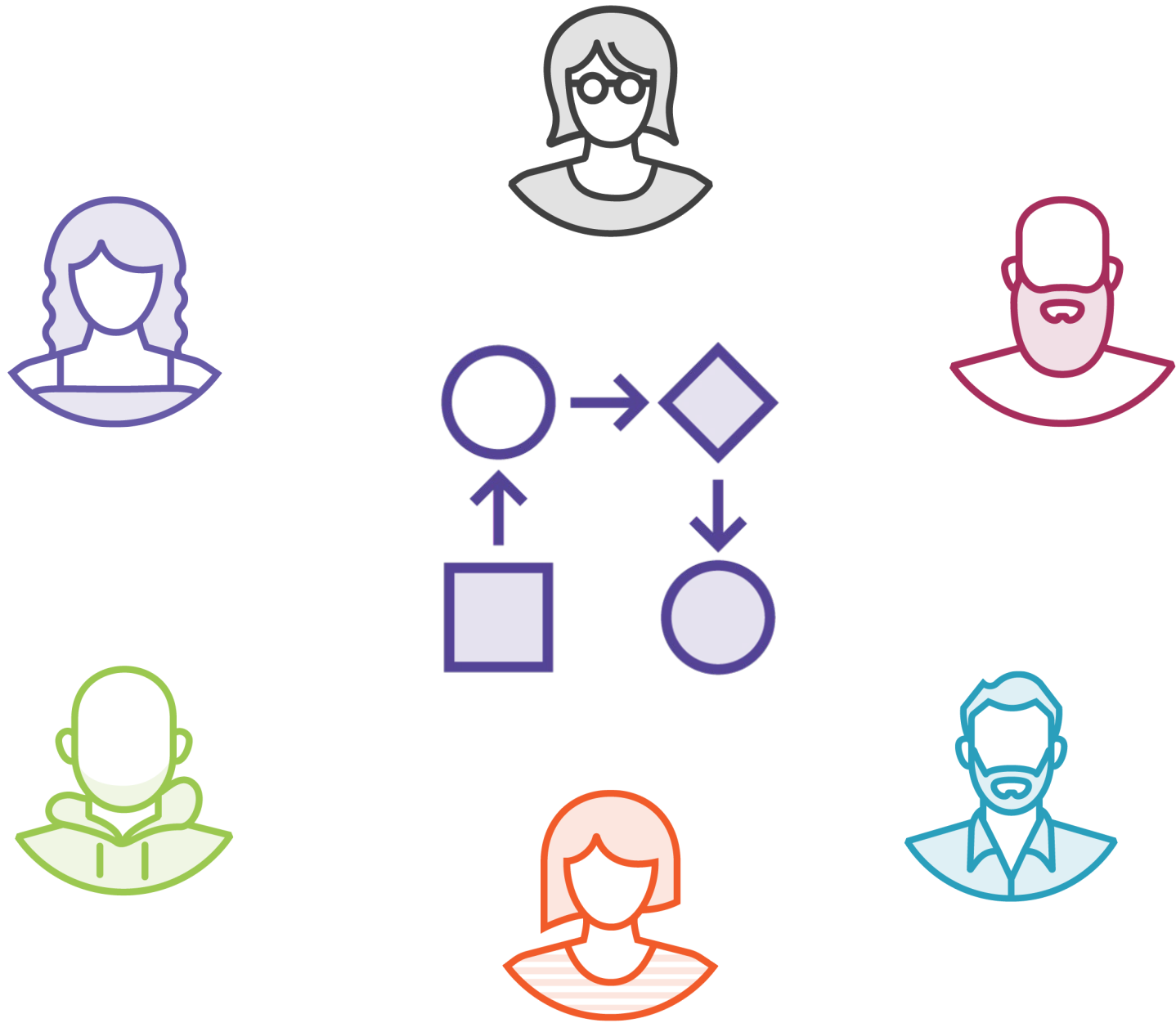


**Aaron Stewart**

PROGRAM ARCHITECT

[www.github.com/a-a-ron](https://www.github.com/a-a-ron)





# git



# GitHub



# GitHub

40,000,000+ developers



Collaboration



Transparency

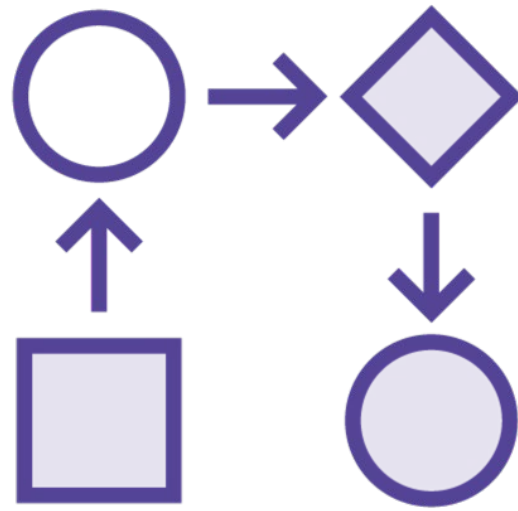


Functionality



Open Source





Workflows



# Overview



**Introduction to Git workflows**

- “Getting Started with Git”

**Learn the GitHub flow**

**Supercharge the GitHub flow**

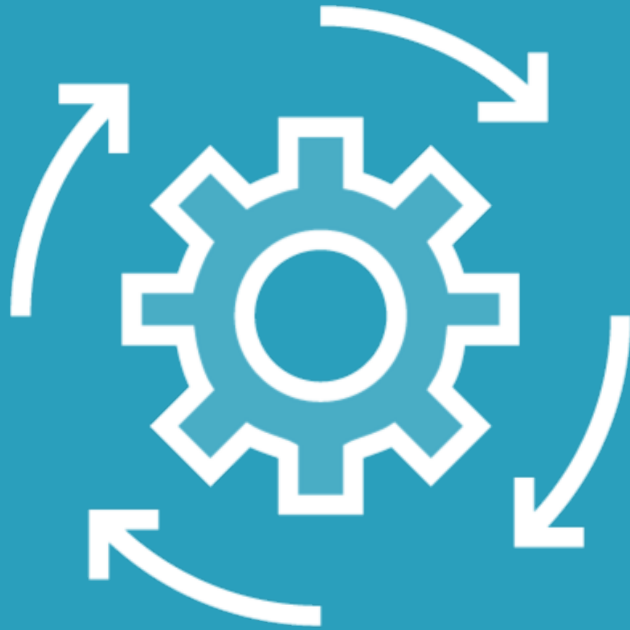
**Merge strategies**

**Customize the GitHub flow**

**Introduction to InnerSource**







Repeatable

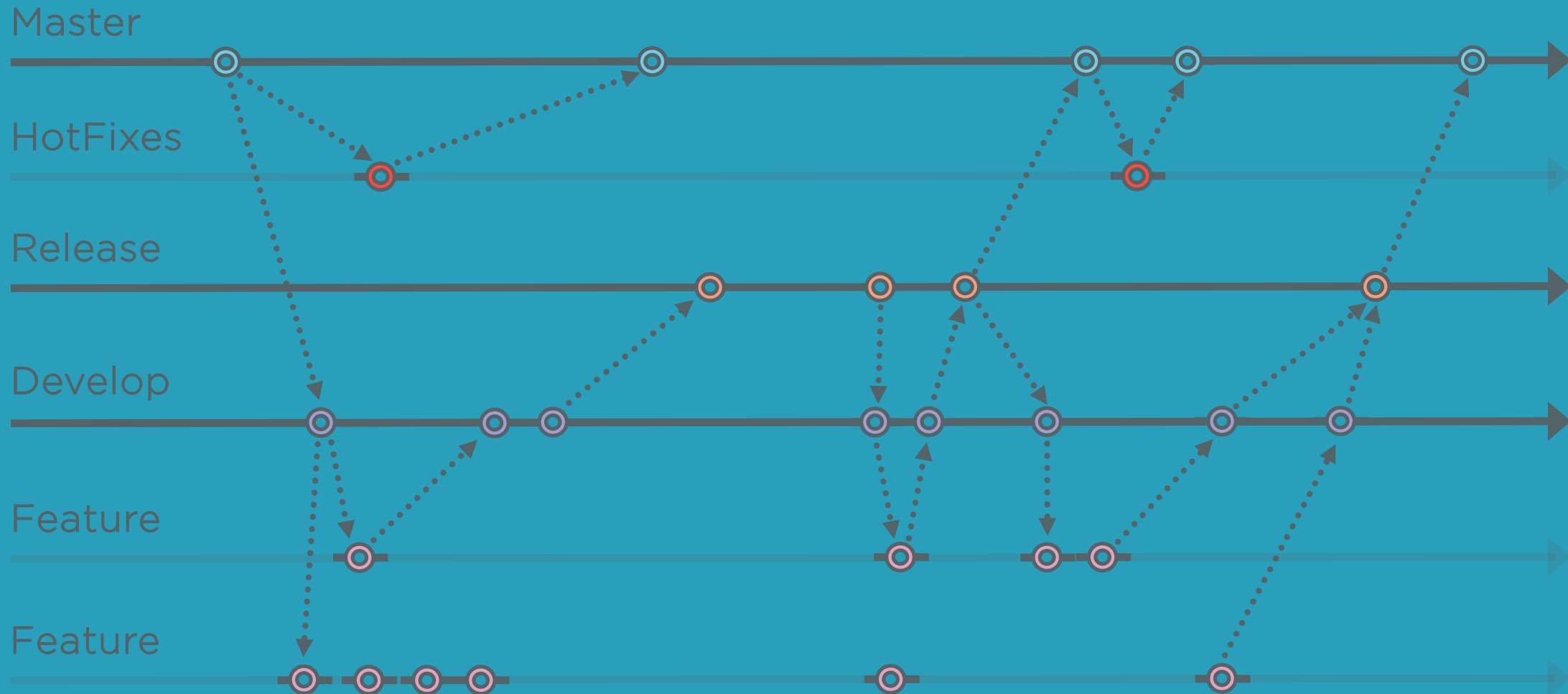


Reviewable





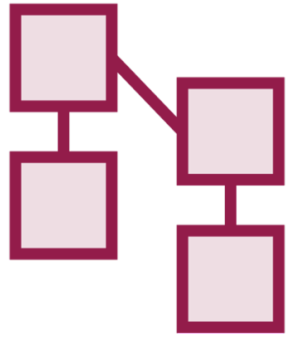
# Git Flow



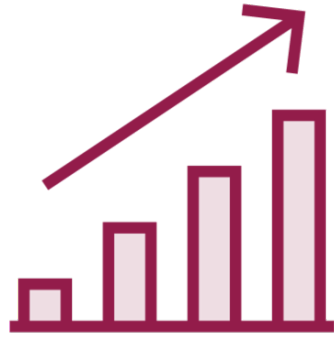
Is there a workflow less complicated and one that allows for **easier collaboration and faster deployments?**







Business



Technology

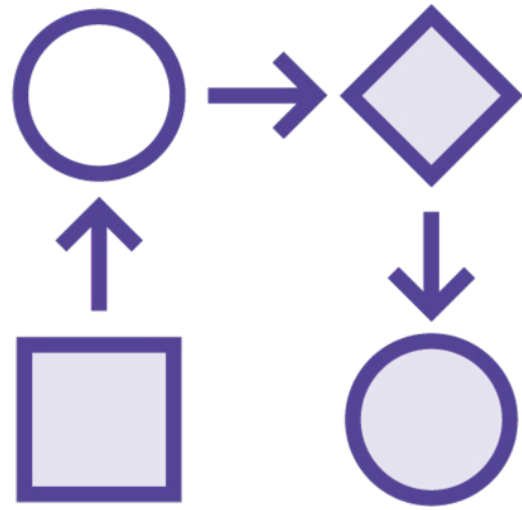


Requirements

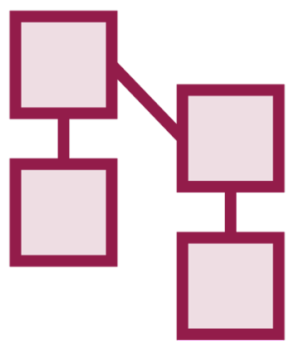


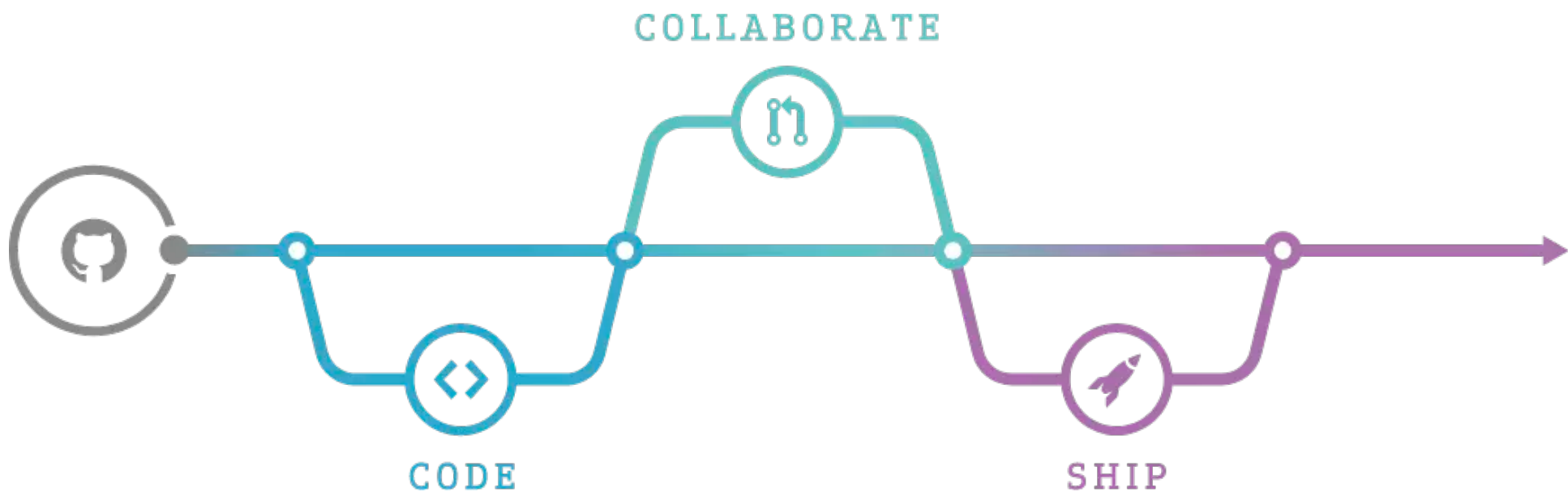
Other ...



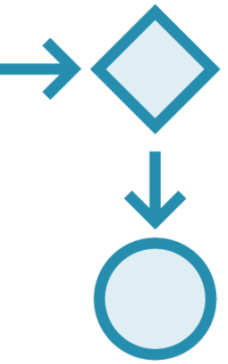


# GitHub Flow

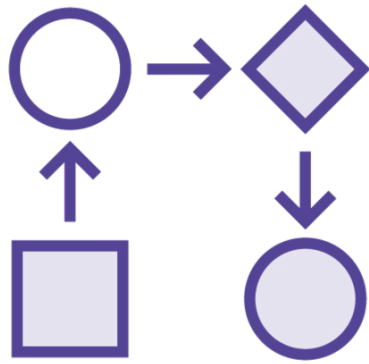




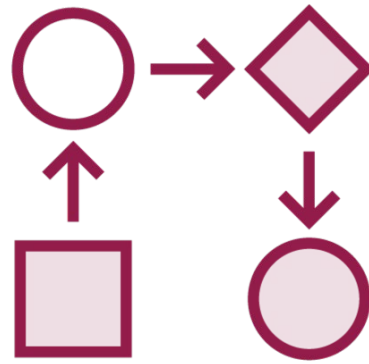
# Possible Git Workflows



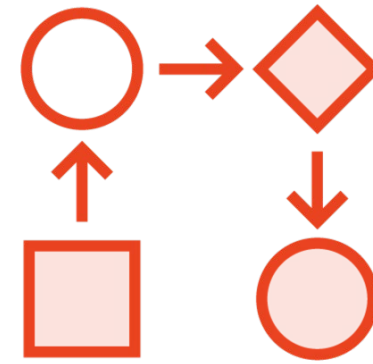
Workflow



Workflow



Workflow



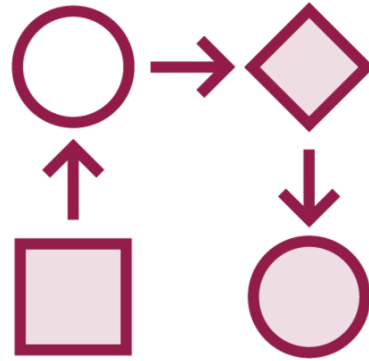
Workflow



Workflow



# Possible Git Workflows



GitHub Workflow



# The Master Branch



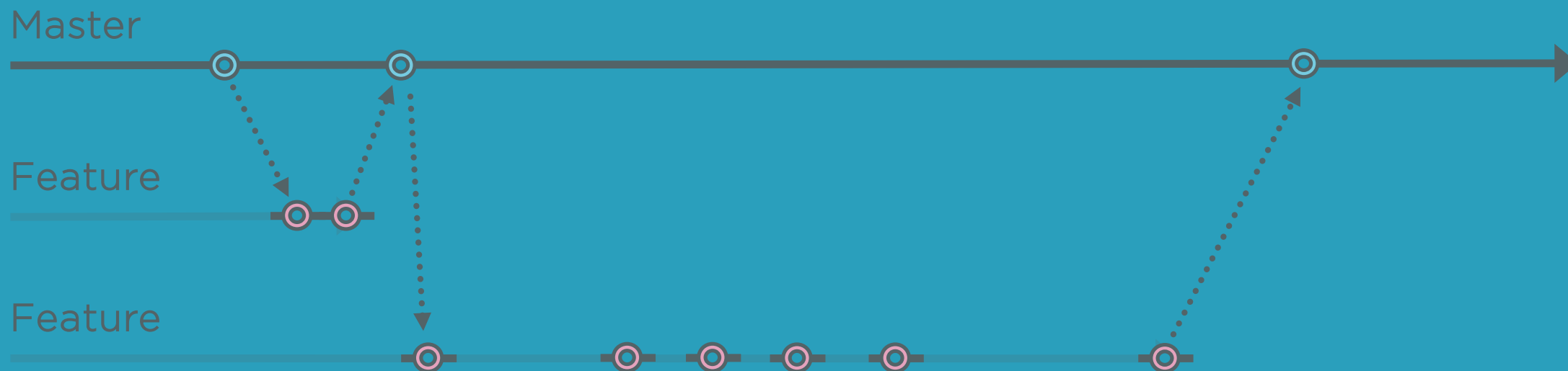
By convention, Git creates “master” as the first branch

Most developers use “master” as the production branch

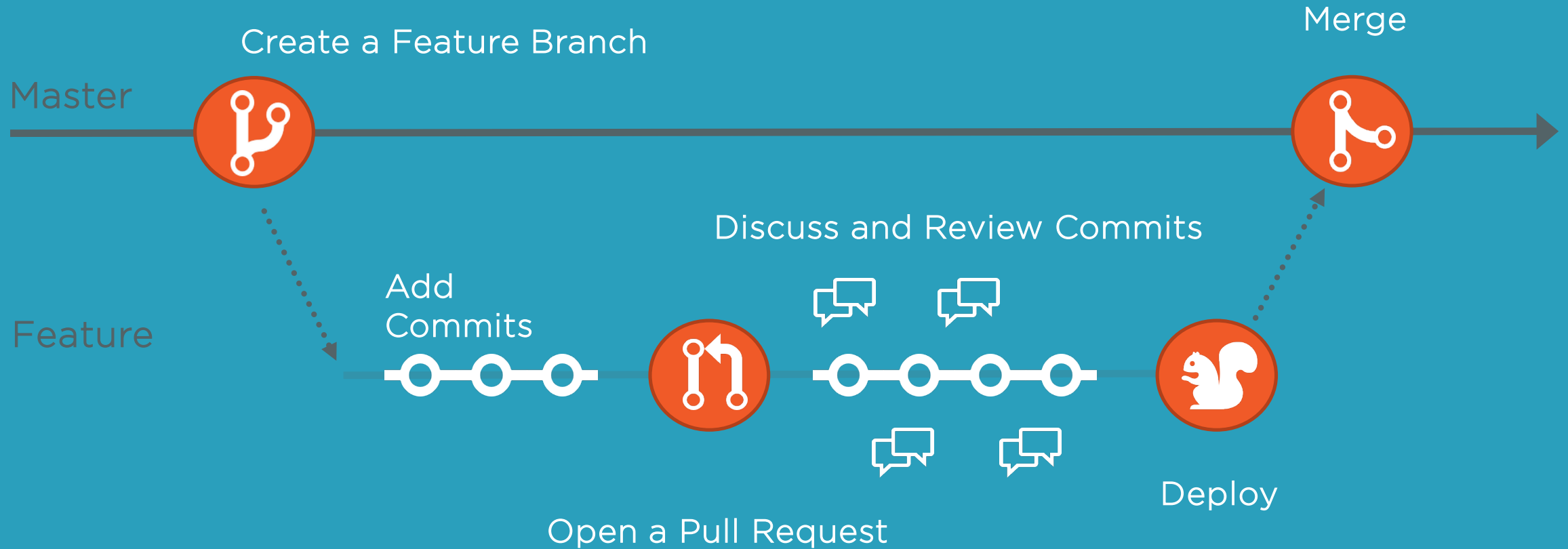
You don’t have to use “master” as the production branch



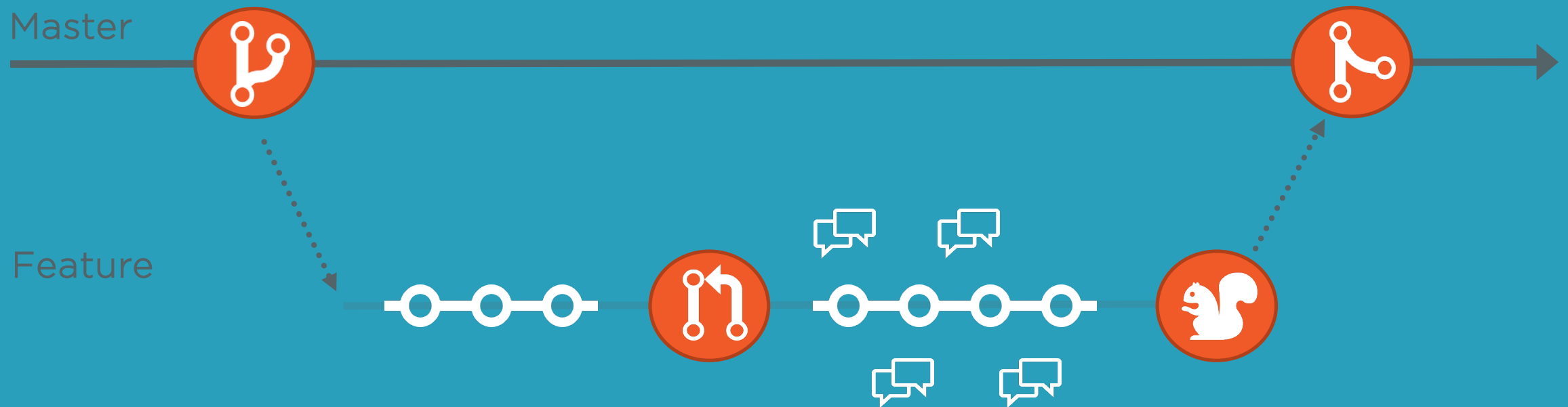
# GitHub Flow



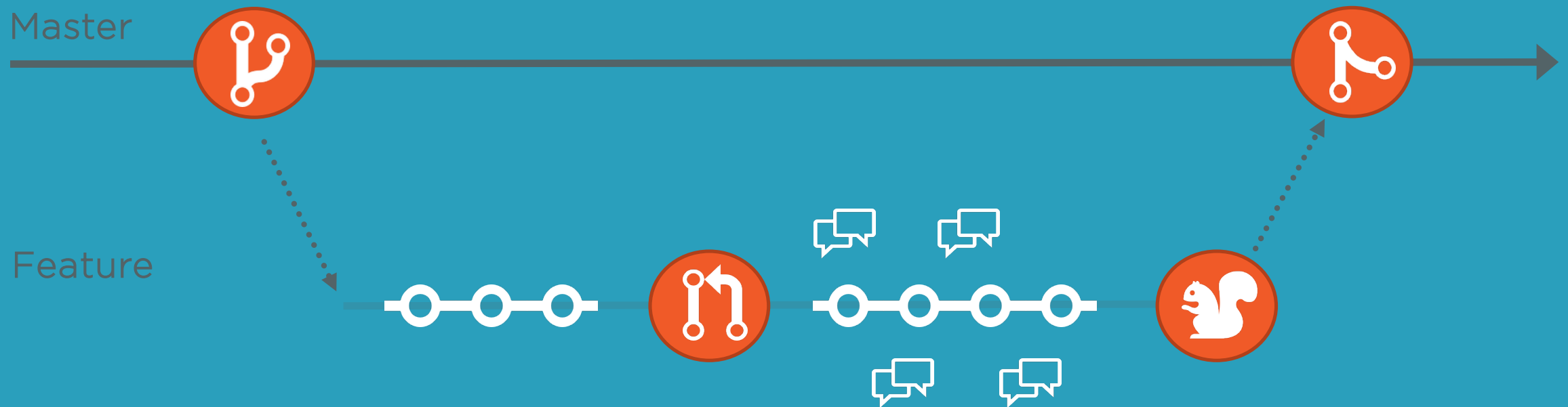
# GitHub Flow



# GitHub Flow



# GitHub Flow



# GitHub Flow



# GitHub Flow



- Security concerns
- Automated testing
- Code reviews
- Other GitHub features

# Securing your Workflow





# Branch Protections





# Protecting The Master Branch

## Rule settings

### Protect matching branches

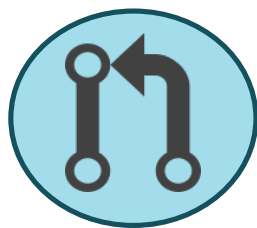
Disables force-pushes to all matching branches and prevents them from being deleted.

**Block direct  
changes made to  
master**

**Only authorized  
users can bypass**

**Can be applied to  
any branch, not just  
master**





# Pull Request Reviews

## ✓ **Require pull request reviews before merging**

When enabled, all commits must be made to a non-protected branch and submitted via a pull request with the required number of approving reviews and no changes requested before it can be merged into a branch that matches this rule.

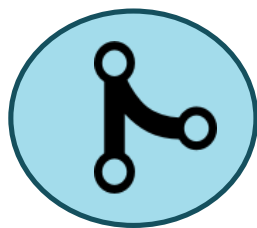
Required approving reviews: 1 ▼

**Require at least one  
approving review**

**Better  
accountability for  
reviewers**

**Better  
accountability for  
pull request authors**





# Merge Protections

☒ **Restrict who can push to matching branches**

Specify people or teams allowed to push to matching branches. Required status checks will still prevent these people from merging if the checks fail.

🔍 Search for people or teams

**People and teams with push access**



**Organization and repository administrators**

These members can always push.

**Comply with  
organization  
requirements**

**Better control over  
who merges**

**Can be authorized  
users or teams**





# Status checks on commits

- ✓ **Require status checks to pass before merging**

Choose which [status checks](#) must pass before branches can be merged into a branch that matches this rule. When enabled, commits must first be pushed to another branch, then merged or pushed directly to a branch that matches this rule after status checks have passed.

Automated control

Consistency with  
each commit

Reduces the amount  
of bugs pushed to  
production



# Continuous Integration

*What is it, and why do I need it?*

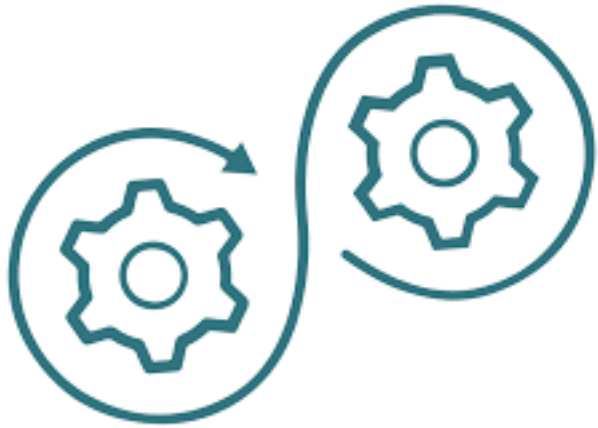


# Continuous Integration (CI)

The process of automating the build and testing of changes when a commit is pushed to a branch



# Continuous Integration (CI)



**Adhere to your team's quality standards**














**Reduce context switching**

**Improve consistency for testing**

**Reduce bugs in production**



# Continuous Integration (CI)

	<b>octocat</b> added some commits 18 hours ago	
	 Platform switcher broken out into own component	6052d0c
	 using JS- classes so we know what is happening	a537a26
	 removing duplicate css	9cd086e
	 cleaning up the content	✗ 5db9a5f
	 Content updates	✗ 786d930
	 Spacing on tips	✓ 52a428d



# Continuous Integration (CI)



**All checks have passed**

4 successful checks

[Show all checks](#)



**This branch has no conflicts with the base branch**

Merging can be performed automatically.

[Update branch](#)

**Merge pull request**



You can also [open this in GitHub Desktop](#) or [view command line instructions](#).

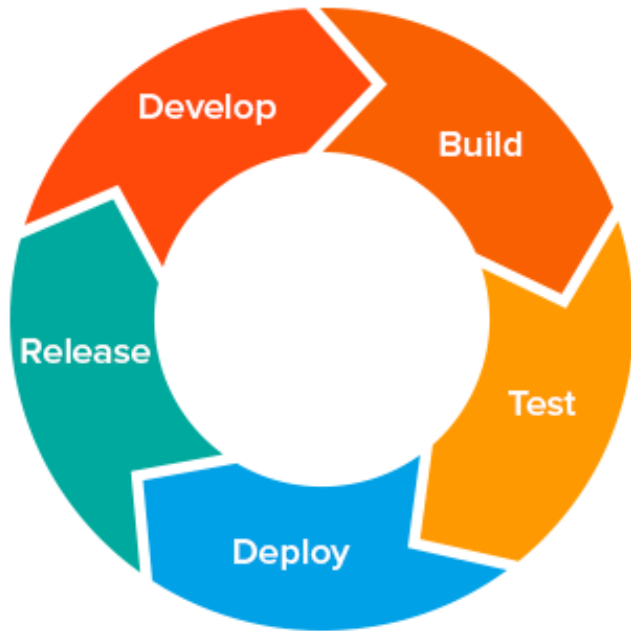


# Continuous Delivery

# Continuous Deployment



# Continuous Delivery



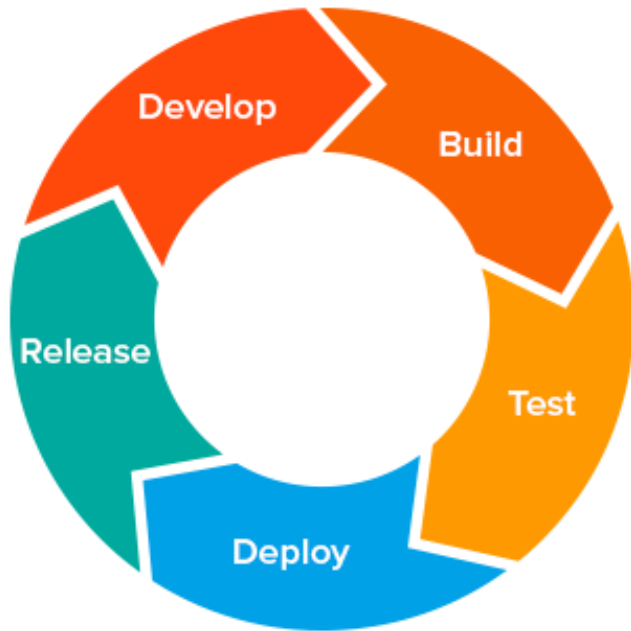
**Production ready with every commit**

**Being production ready doesn't always translate to releasing new code every time**

**Gives you the flexibility to release as often as you want**



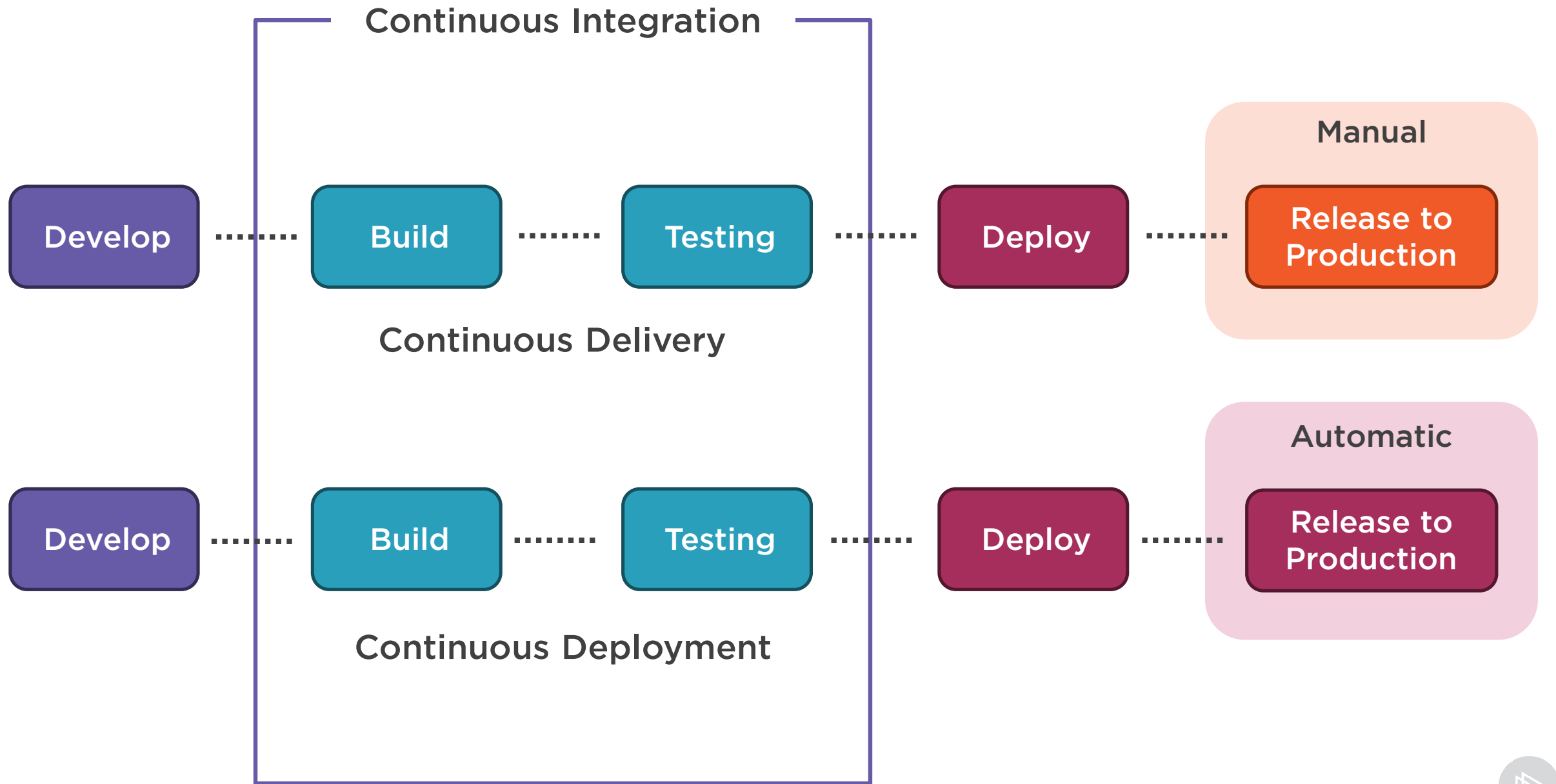
# Continuous Deployment (CD)



**Takes continuous delivery one step further**

**Automated in the background, no human intervention**

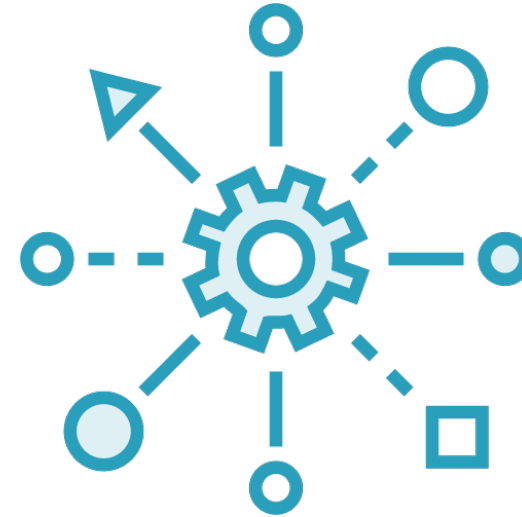




# Continuous Deployment



Deployment releases to production are made often

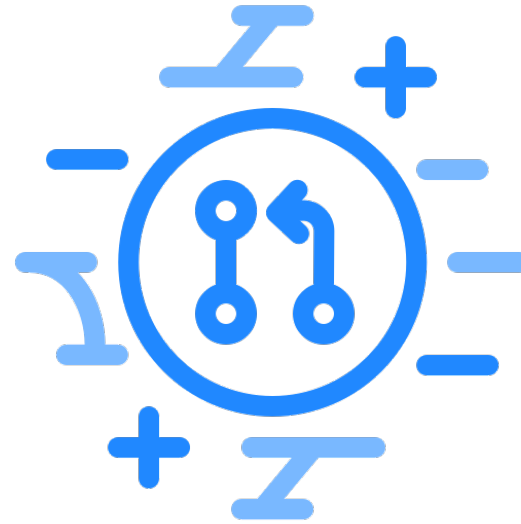


Closes the gap in customer feedback, feature additions, and bug fixes





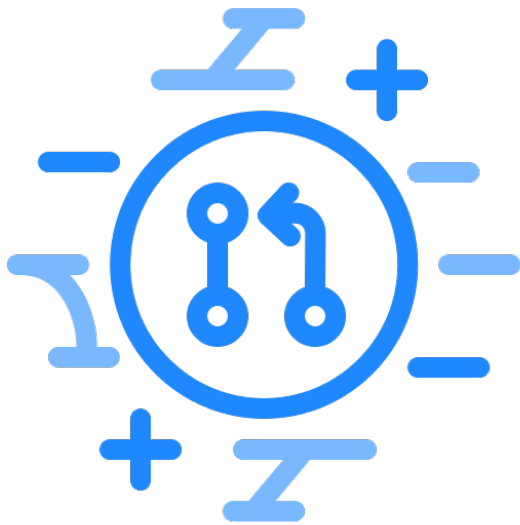
Issues



Pull Requests



# Pull Requests



**Introduce collaboration and visibility around the changes you're making**

- Discuss and review changes
- Allow you to modify, test, and confirm changes in a safe, non-destructive environment.

**When to open a pull request?**

- Changes don't need to be perfect
- Does fear or culture affect this?

# GitHub Flow



# Issues

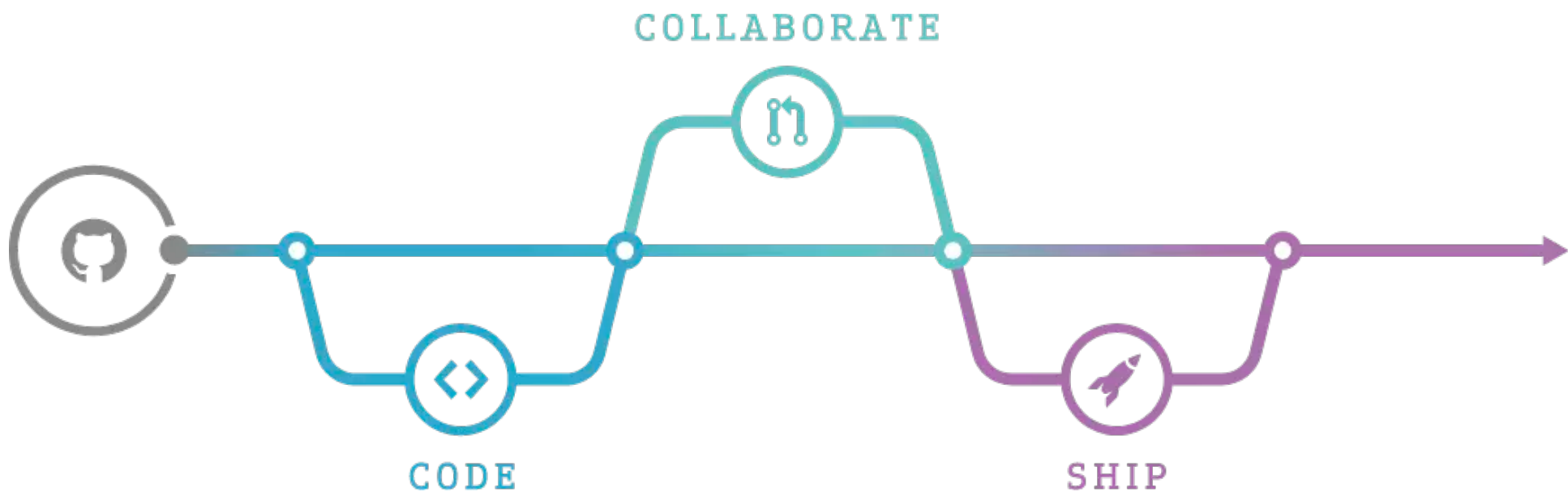


**Another GitHub feature to add collaboration to your project**

**Discussion based**

- New features, direction, hackathons, project planning, etc

**Provides flexibility on how to use them based on your project's needs and workflow**



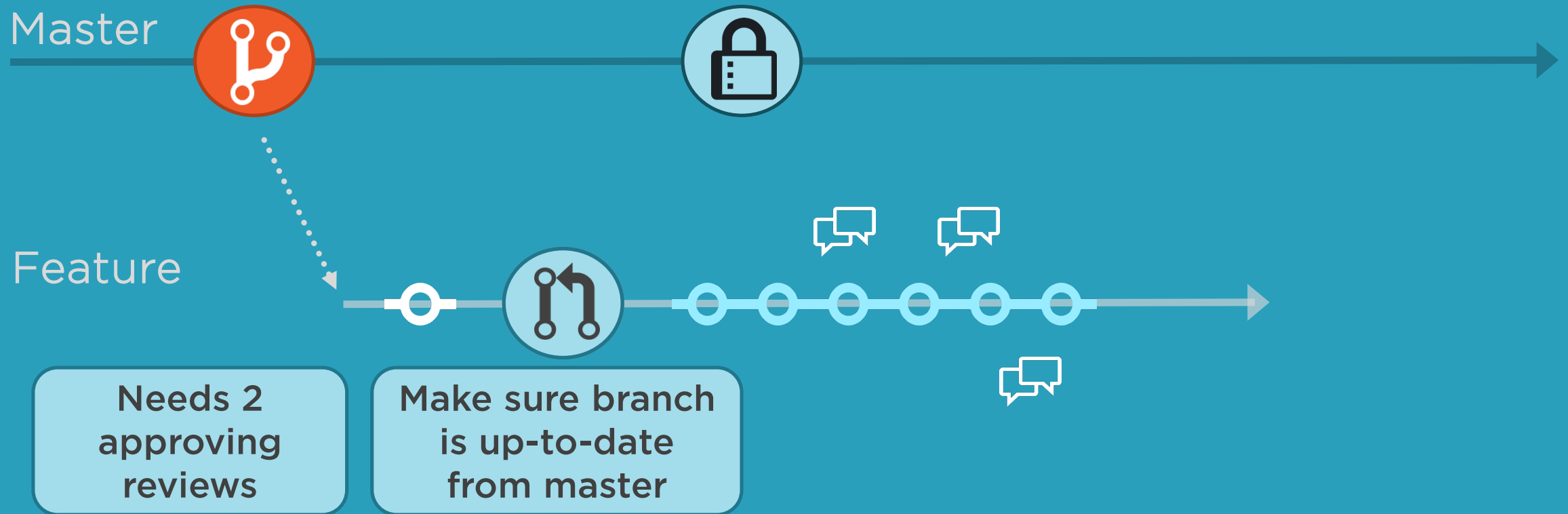
# Supercharged GitHub Flow



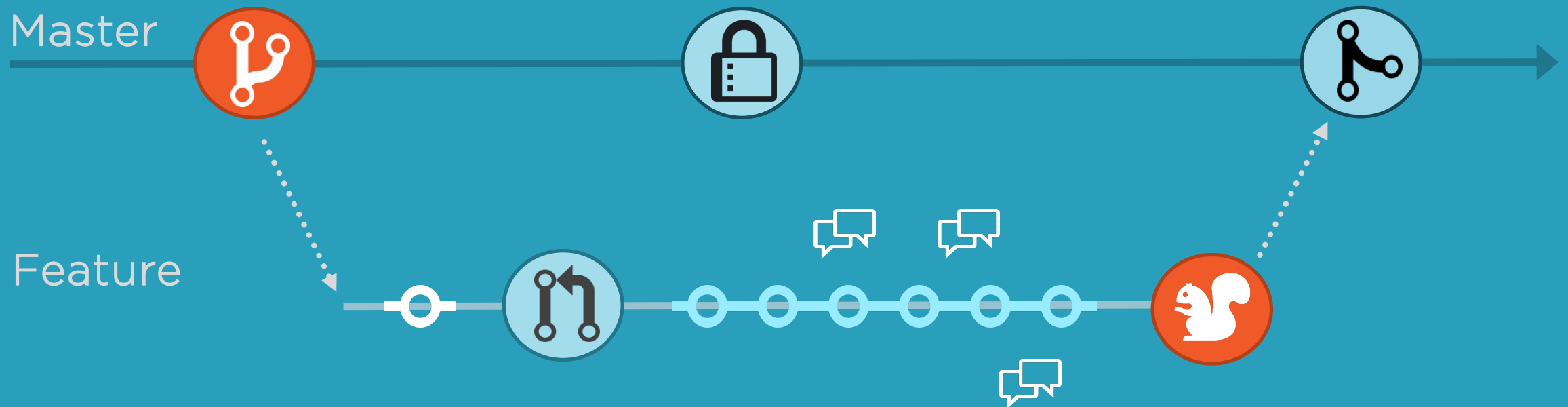
# Supercharged GitHub Flow



# Supercharged GitHub Flow



# Supercharged GitHub Flow

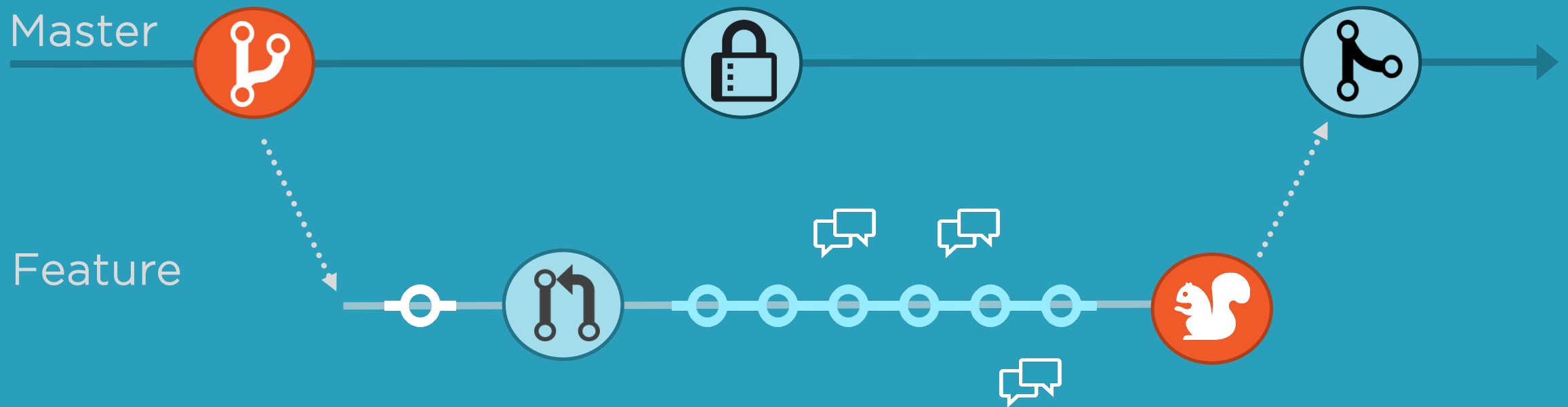


Only those approved can  
merge





# Supercharged GitHub Flow



# Summary



Git flow

GitHub flow

Securing your workflow

Continuous integration

Continuous delivery/deployment

Issues/pull requests

How it all fits together

