

Customizing Your GitHub Workflow



Aaron Stewart

PROGRAM ARCHITECT

www.github.com/a-a-ron

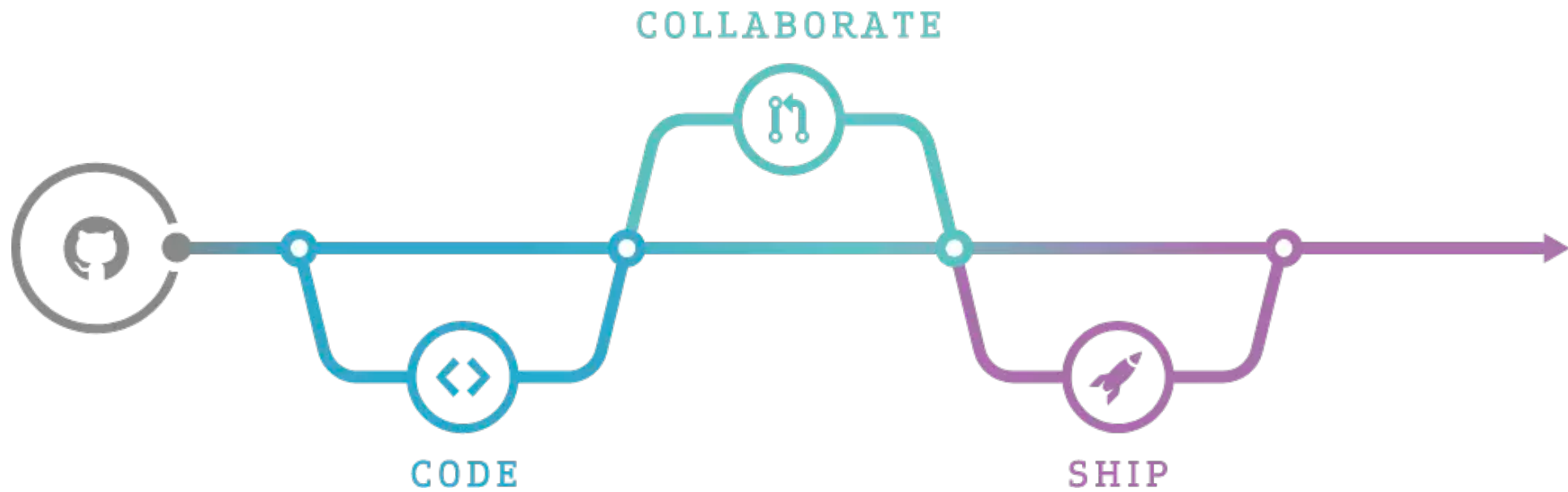


This is great, but we work a little
bit differently





GitHub Flow



Up to this point

Security features

Code reviews

**Continuous
integration**

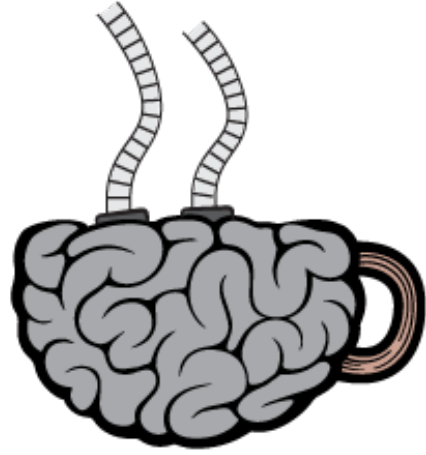
**Continuous
delivery**

**Continuous
deployment**

**Issues
Pull requests**





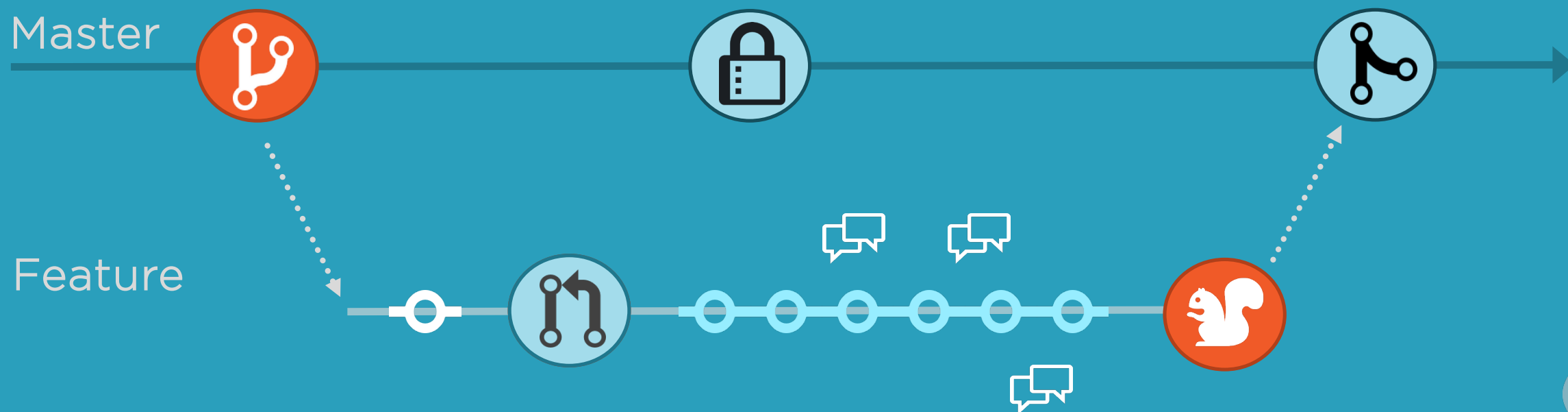


WIRED BRAIN
— COFFEE —





GitHub Flow



Merge strategies

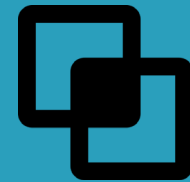




Thinking about Merge Strategies



Fast Forward

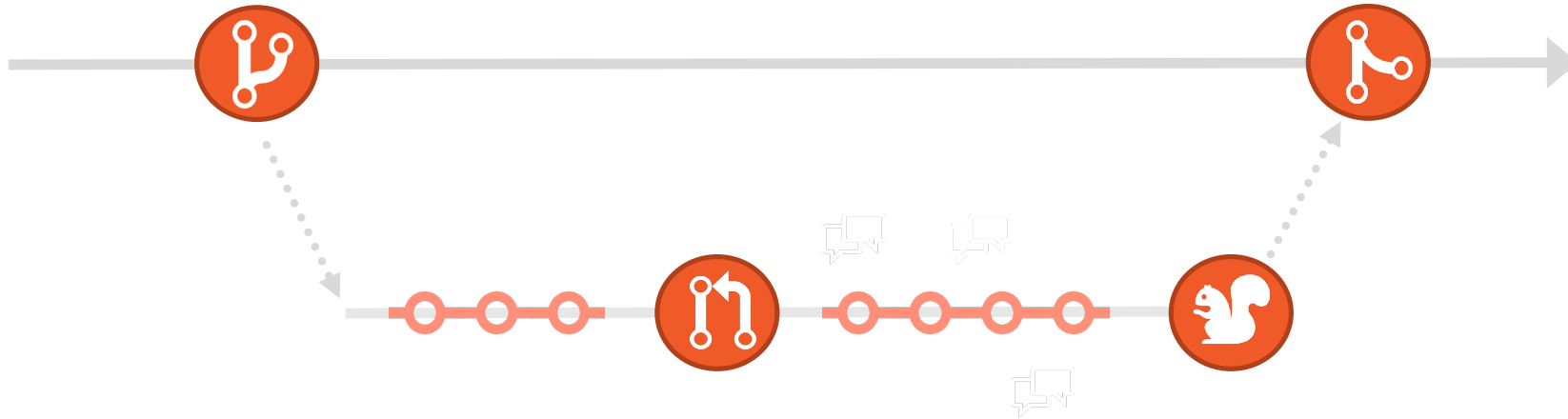


Recursive





Fast Forward Merge

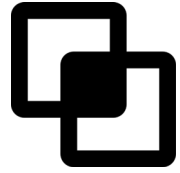


**No new commits on
the base branch**

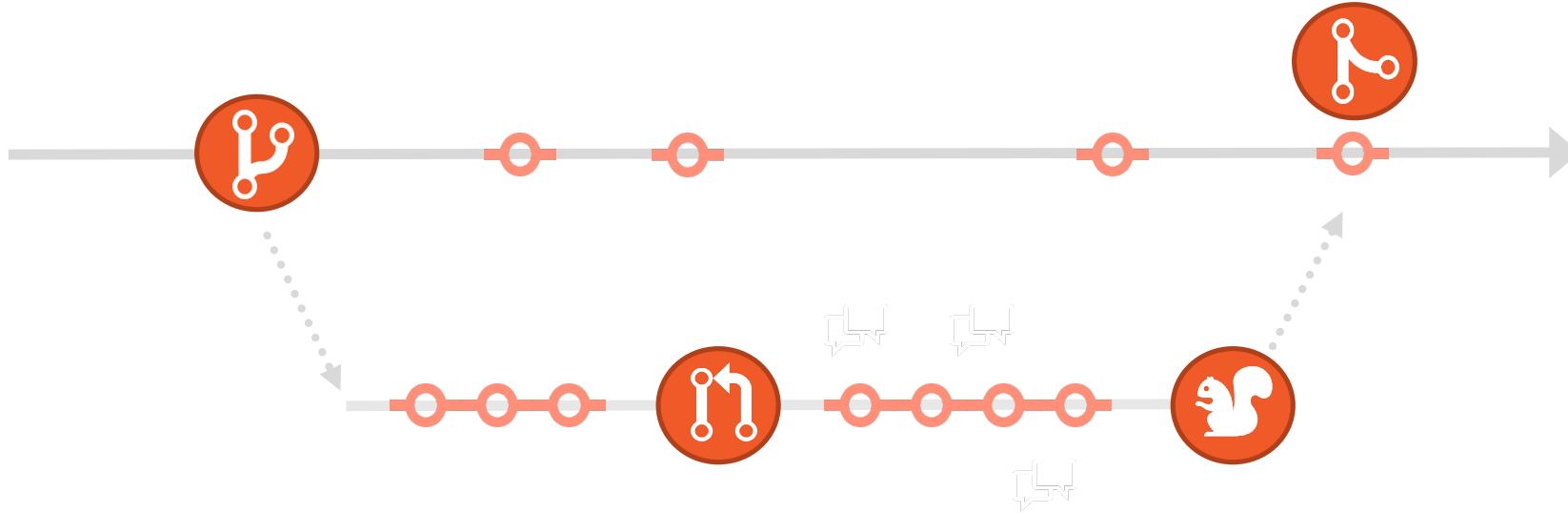
**No merge commit
on the base branch
when merged**

**Provides a linear
history**





Recursive Merge



New commits on
the base branch

A merge commit is
created on the base
branch

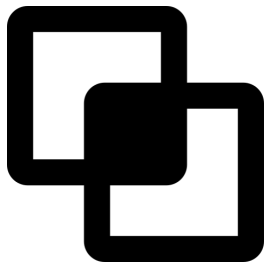
Merge commit has 2
parents with a true
diverged history



Which One Should I Use?



Or



Fast Forward Merge

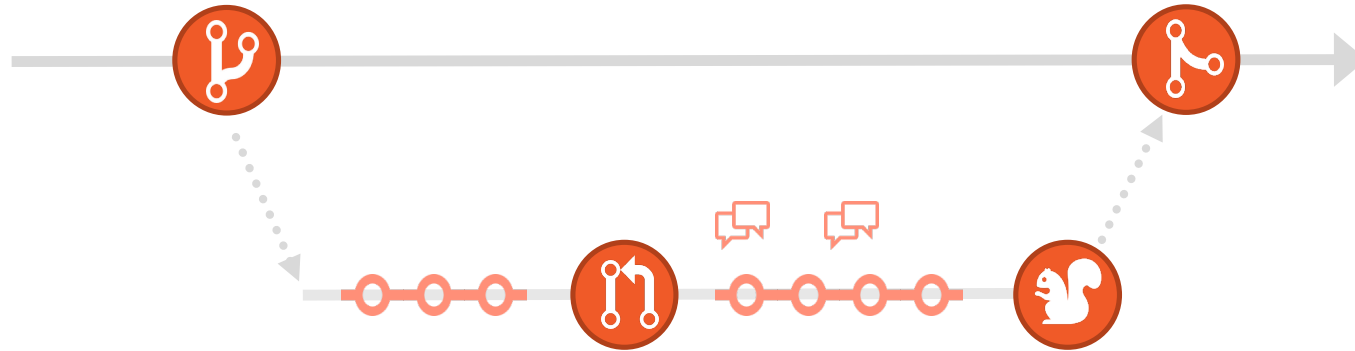
- Use the ``git rebase`` command
- Provides a linear history with no merge commit

Recursive Merge

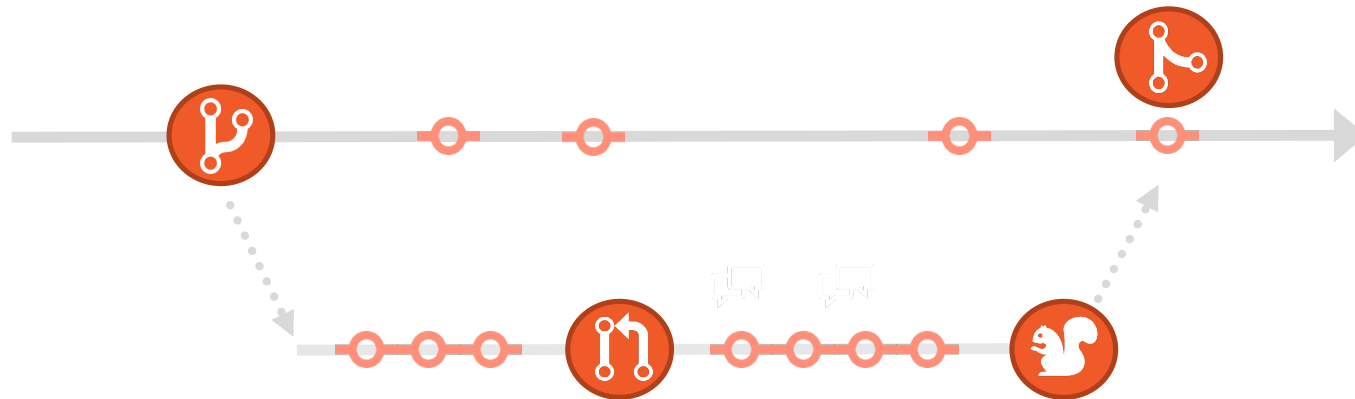
- Use the ``git merge --no-ff`` command
- Provides a merge commit with a true deviating history

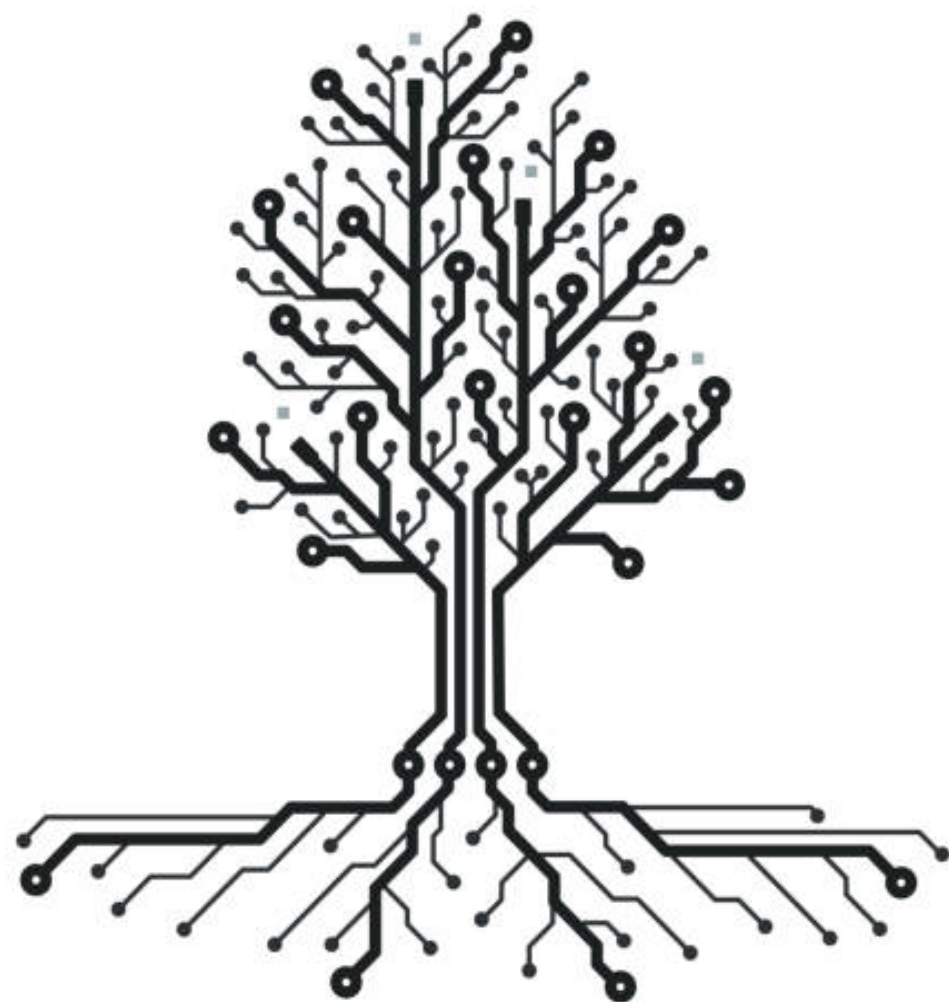


Fast Forward Merge

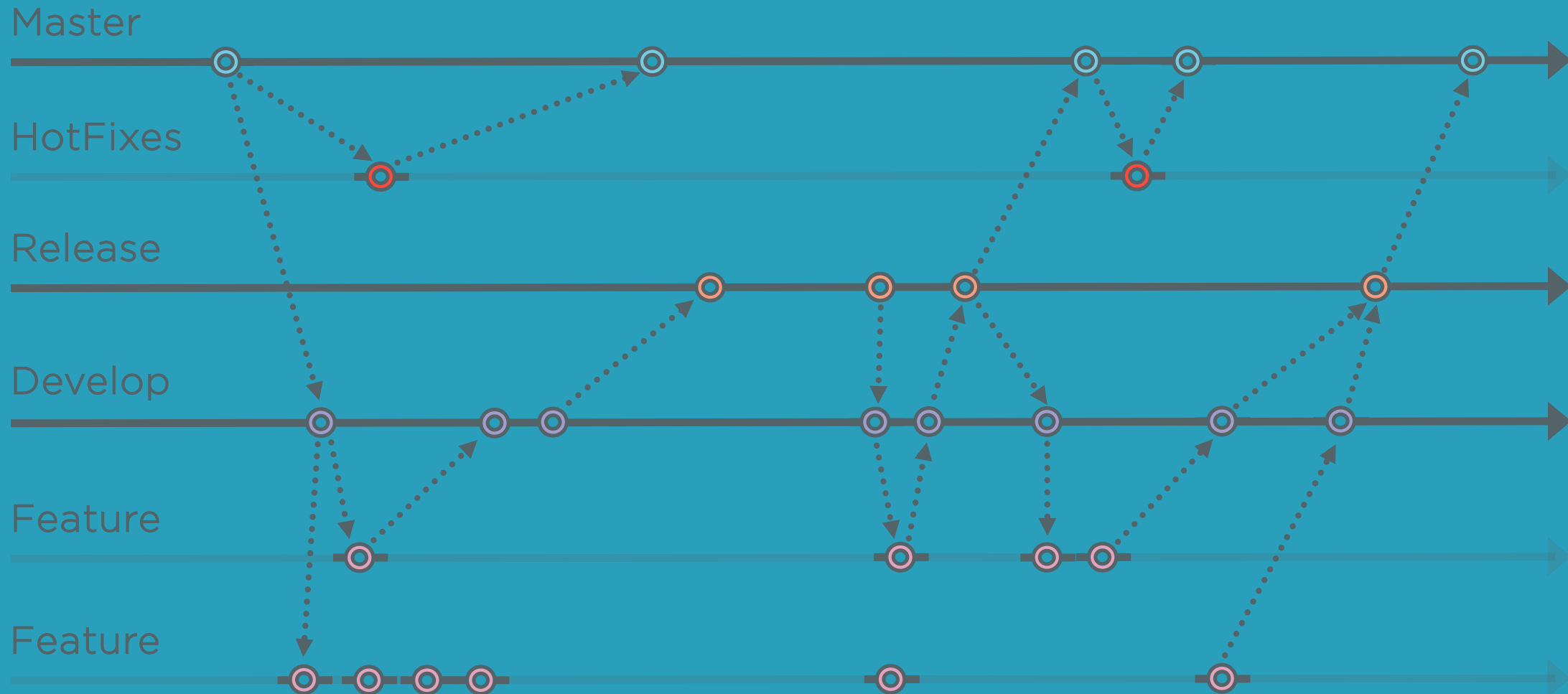


Recursive Merge





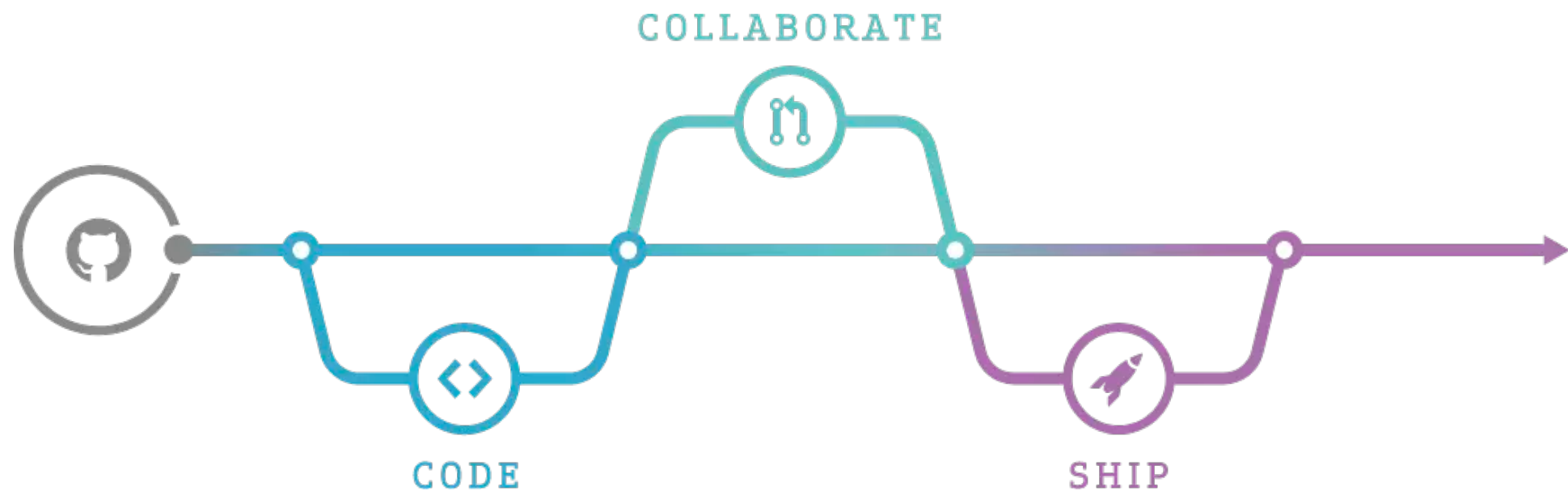
Git Flow



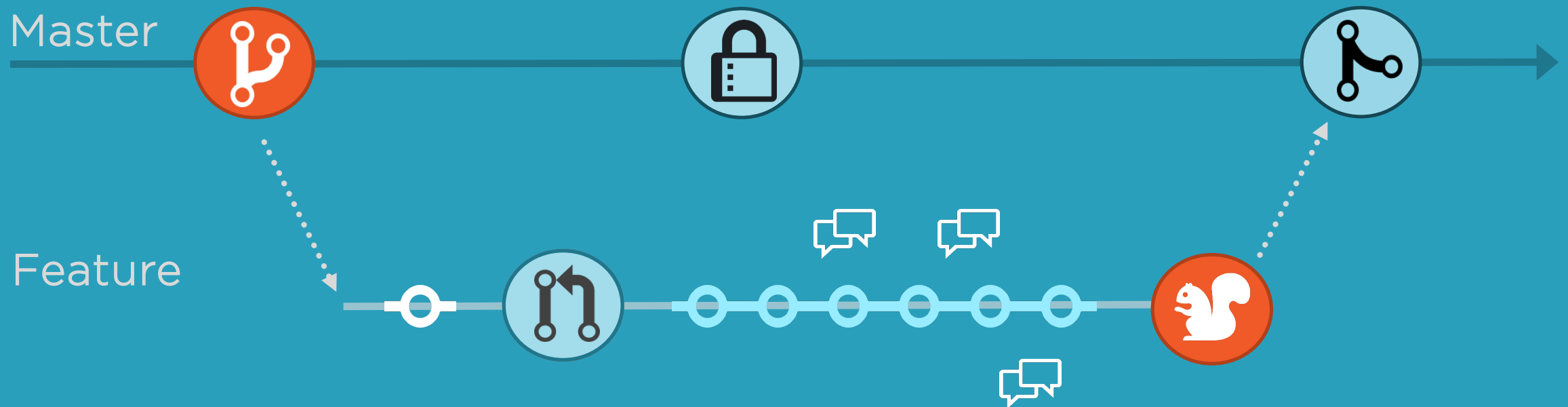


Why are we going back to the Git
flow?

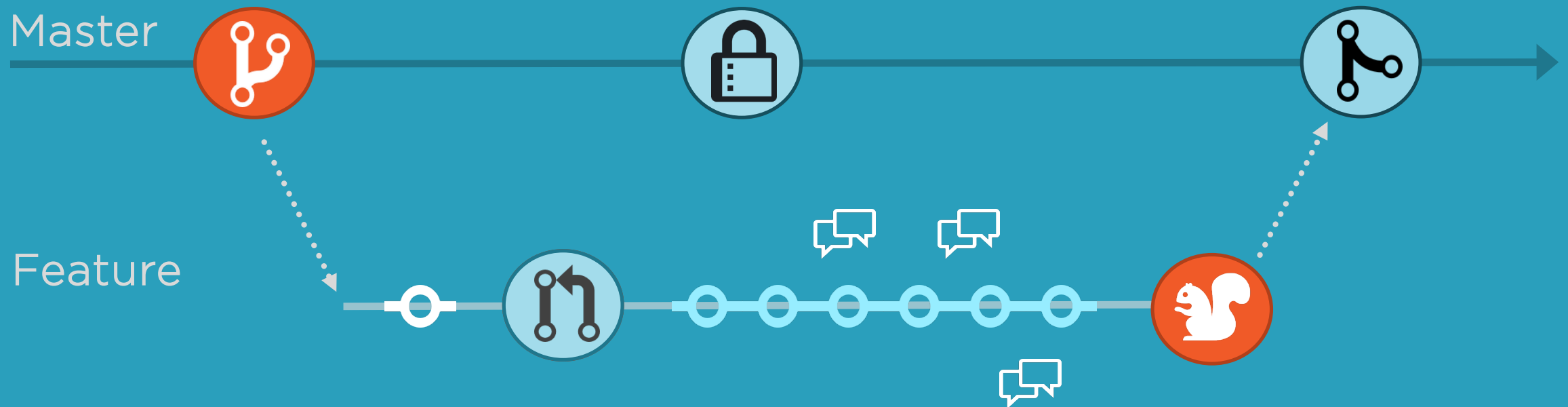




GitHub Flow

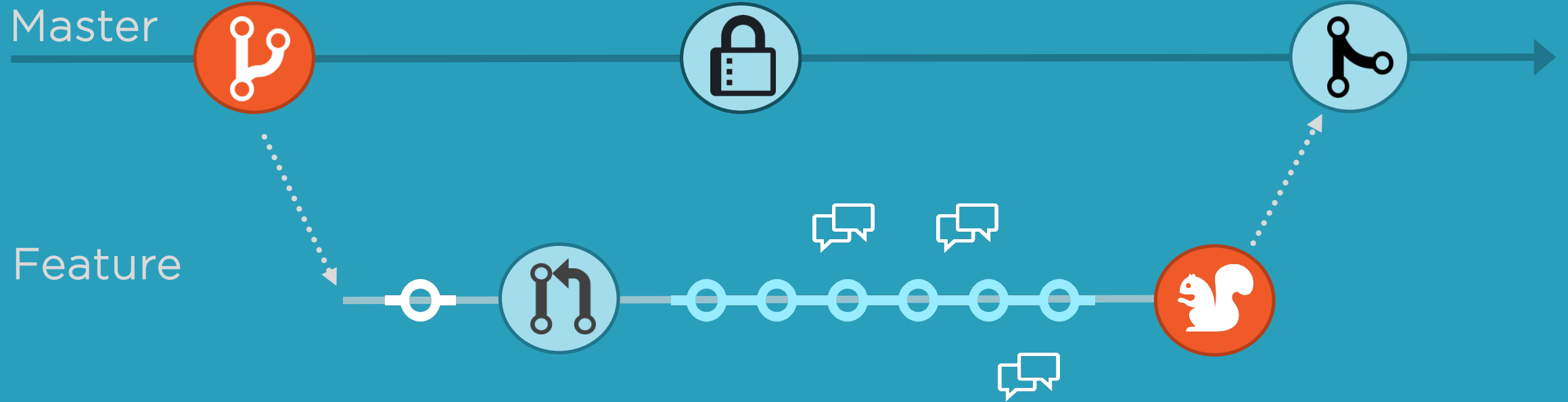


GitHub Flow



It's too simple. Our teams need
more control.





Master

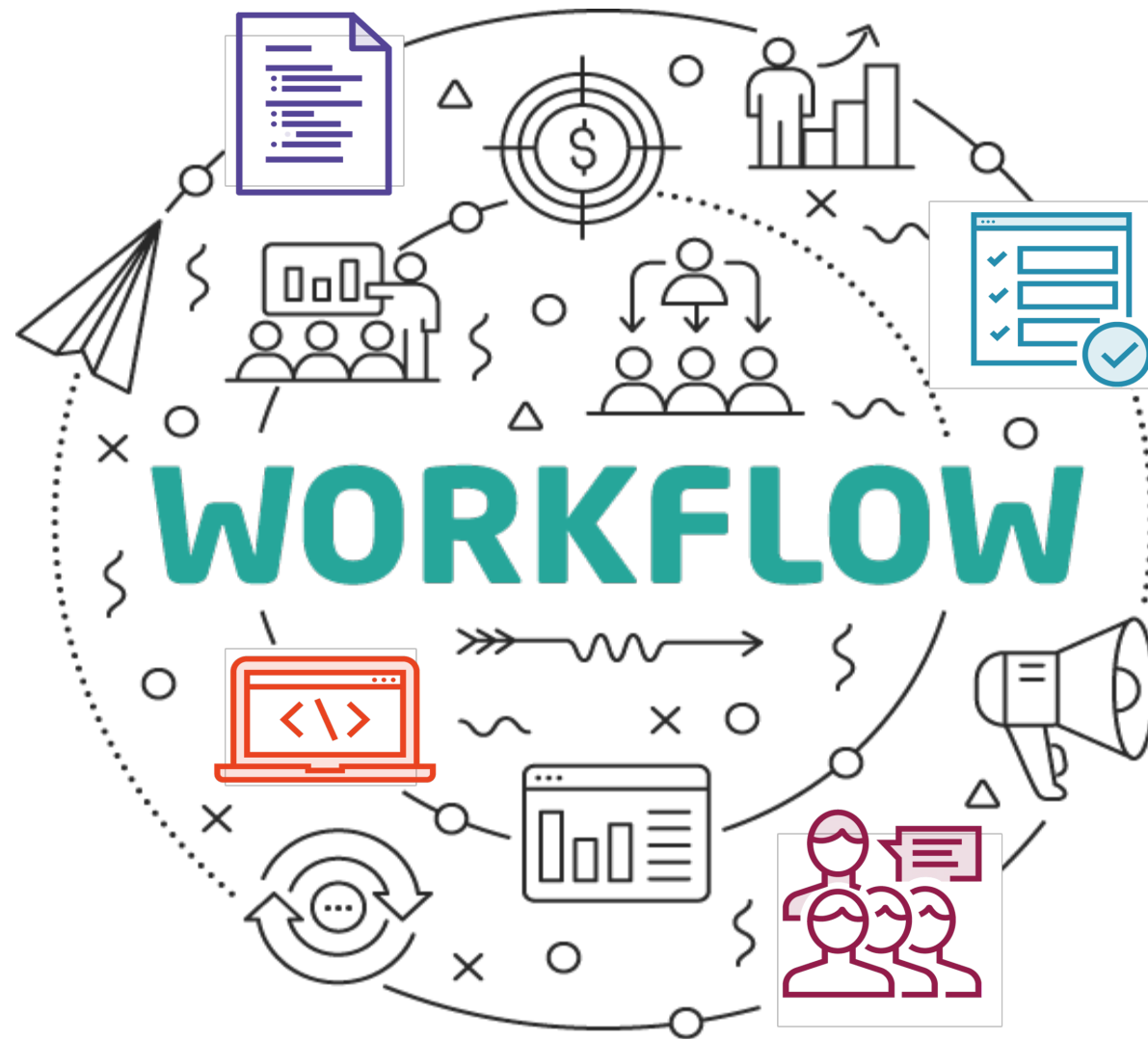


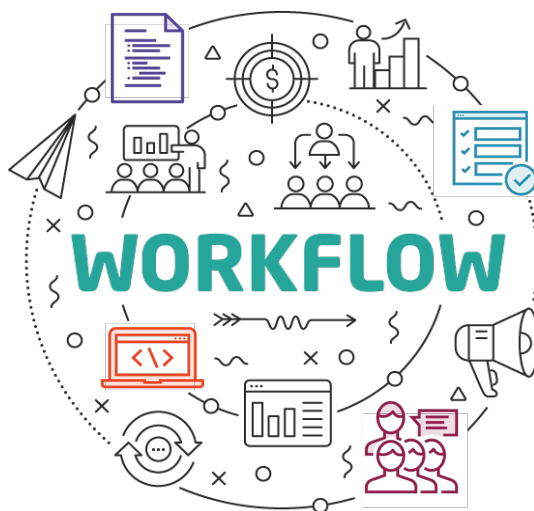
Develop

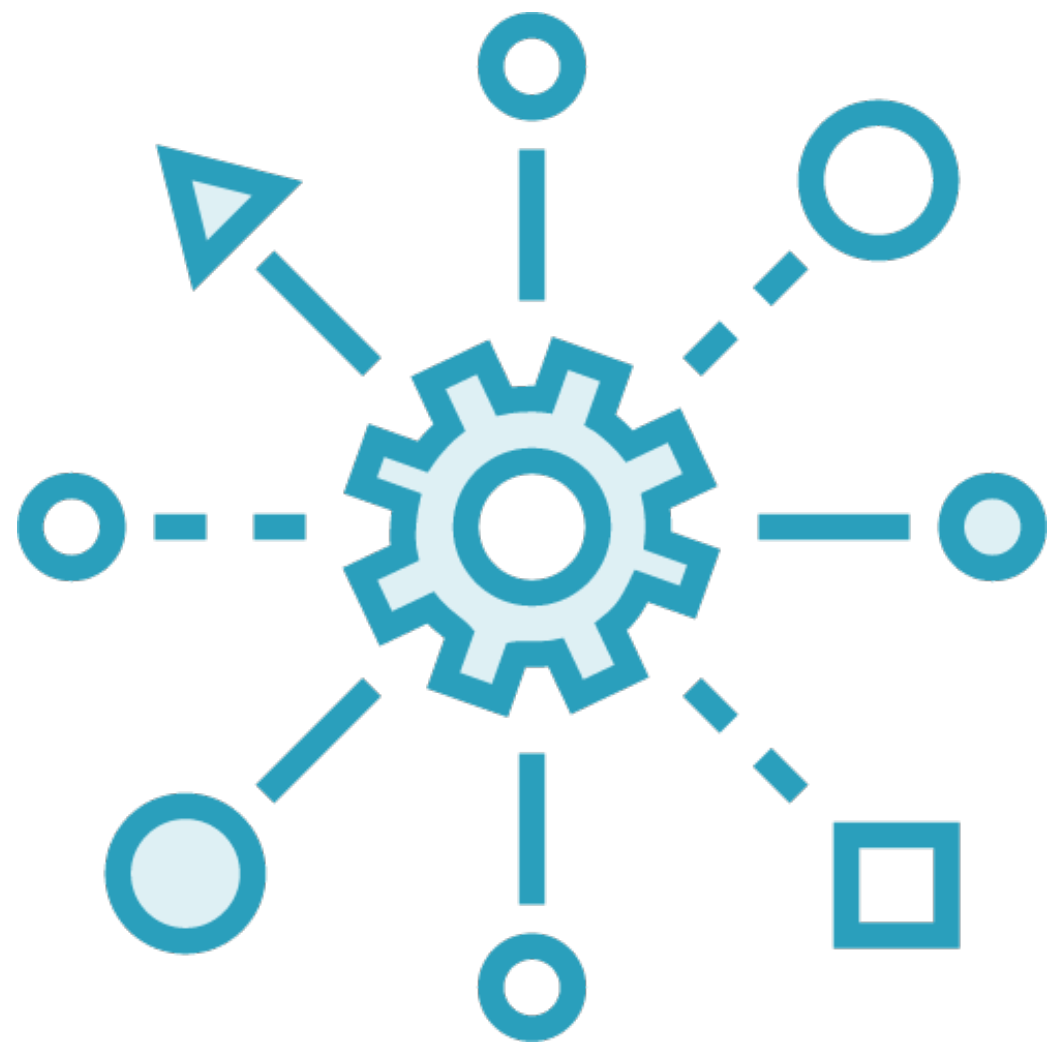


Feature









Workflow Considerations

Where do we go from here?



Considerations for Your Workflow



Which branching strategy will you use?

Which branch will serve as your “master” branch?

How will you protect your code?

Will you use naming conventions for your branches?

How will you use issues and pull requests?

Considerations for Your Workflow



How will you indicate sign-off on pull requests?

How will you teach your workflow to your team?

What integrations will be used?

If users have questions about Git, GitHub, or their workflow, who do they ask?



Workflow Challenges

Are the challenges just technical?



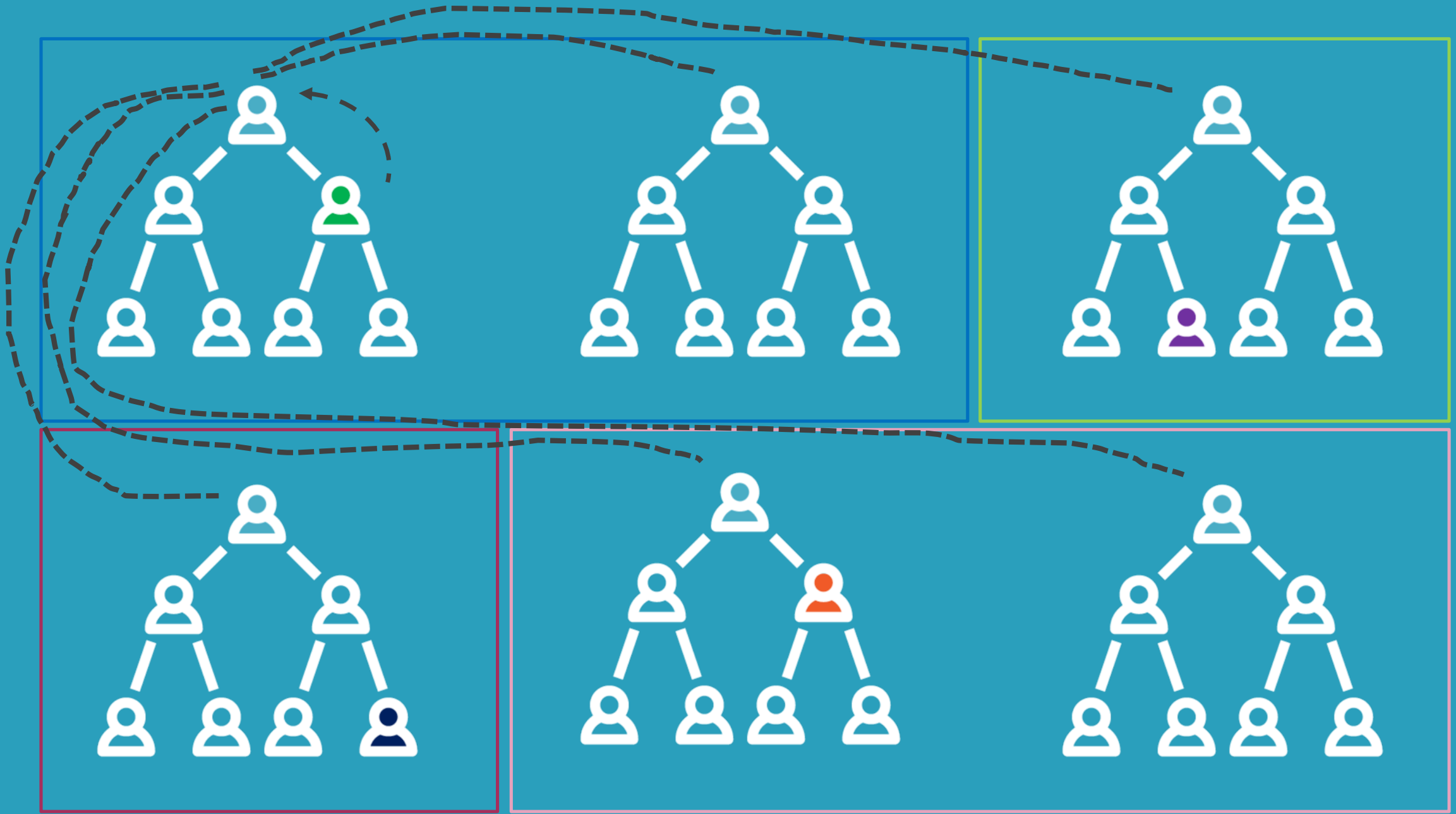
Classic Vertical Organization

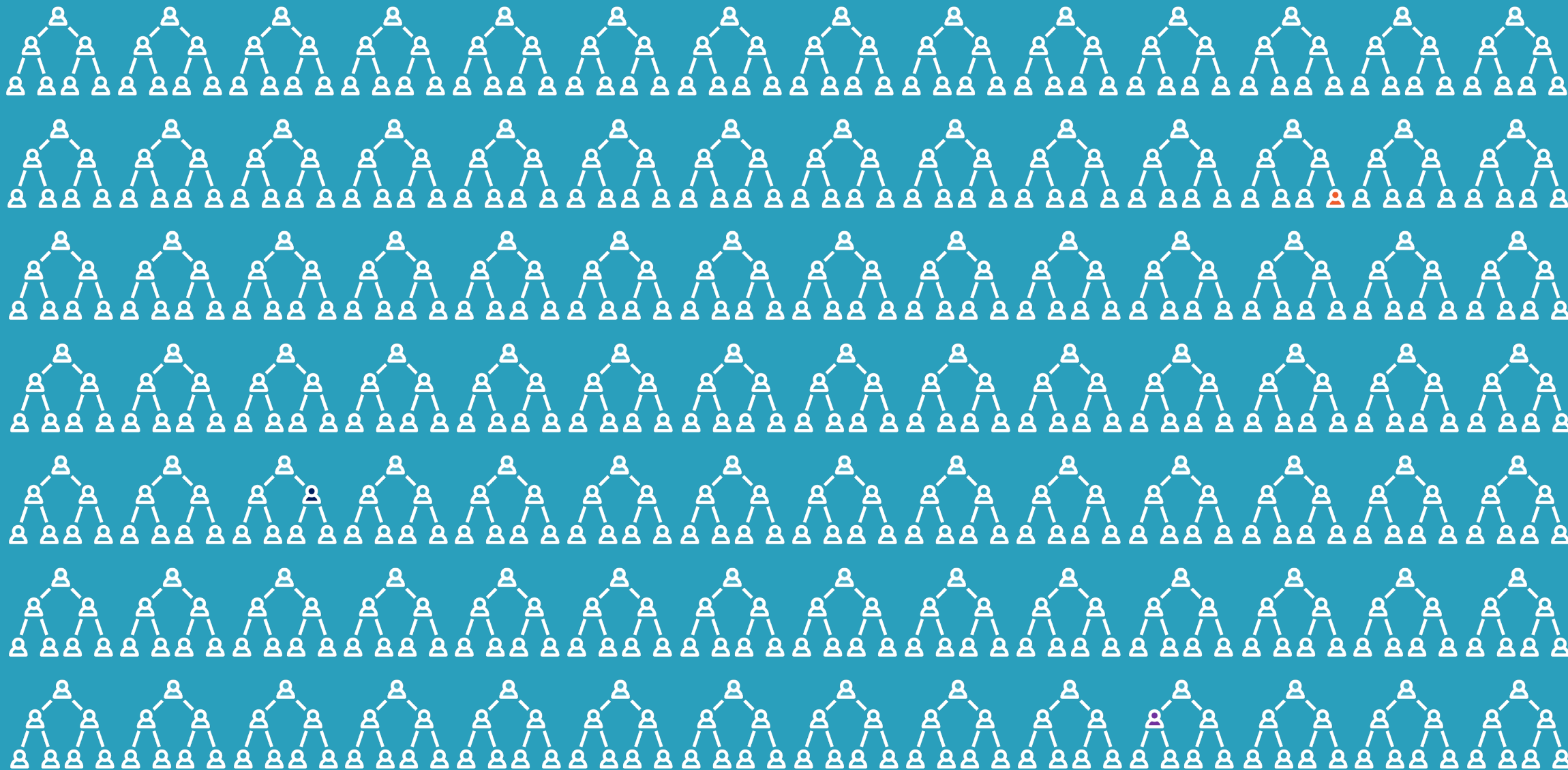


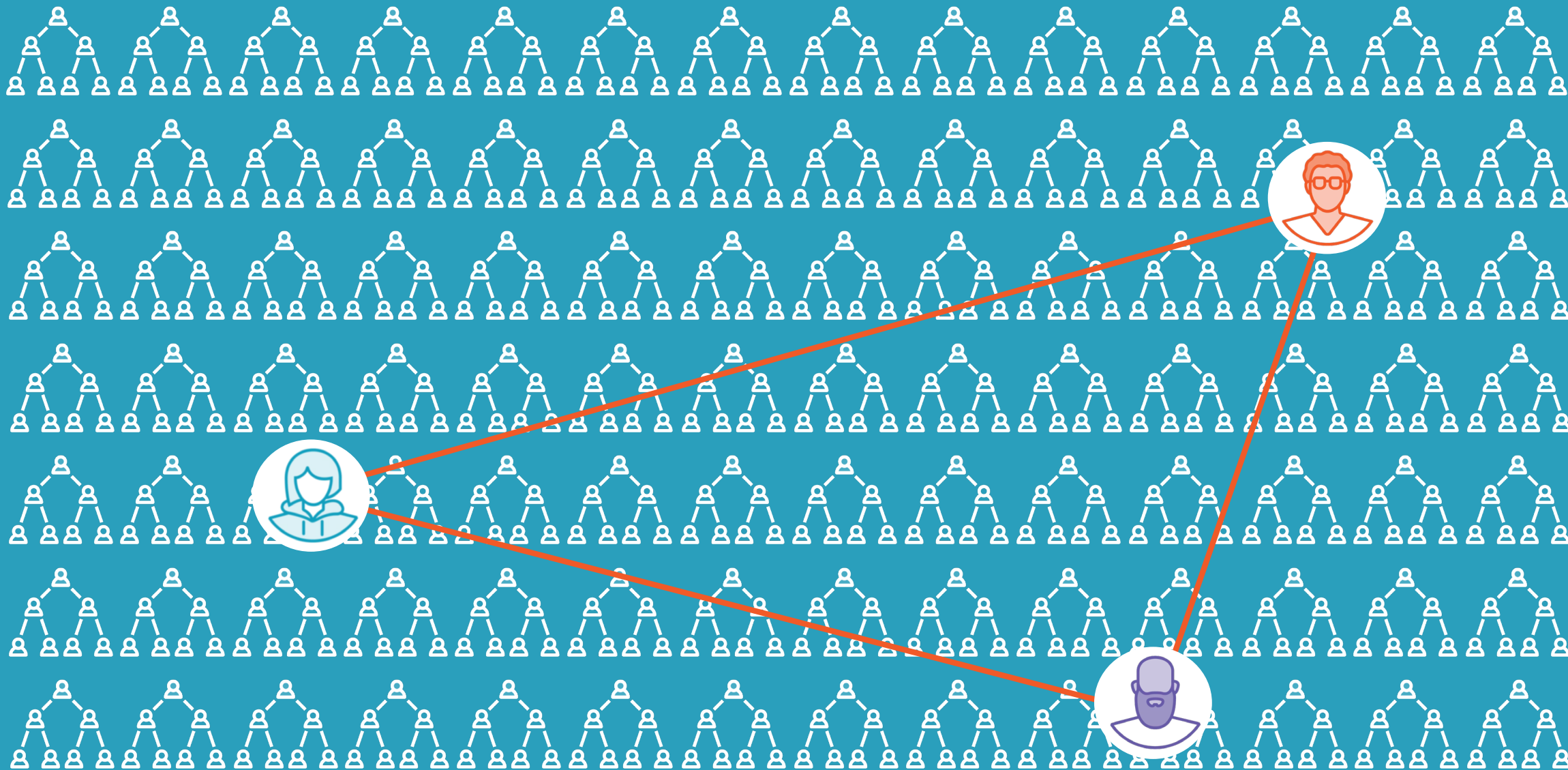


Most Organizations struggle with
awareness of existing enterprise
development









Creating an Open Culture through InnerSource



InnerSource

Bring the successful practices of the open source community inside the four walls of your organization. Practices like increasing transparency, trust, and collaboration are core to innersource.



Adopting an InnerSource Culture



Open collaboration encourages more contributions

Developers don't always have to start from scratch

Transparent decision-making builds process, trust, and alignment

Higher quality development and better documentation





Summary



Use the GitHub flow as your foundation,
and customize as needed

Have those important discussions

Workflow and culture change doesn't
happen overnight

Adopt an InnerSource culture



Thanks for watching, and
happy learning!

