Branching Strategies For Every Occasion

Xavier Morera

PASSIONATE ABOUT TEACHING

@xmorera www.xaviermorera.com

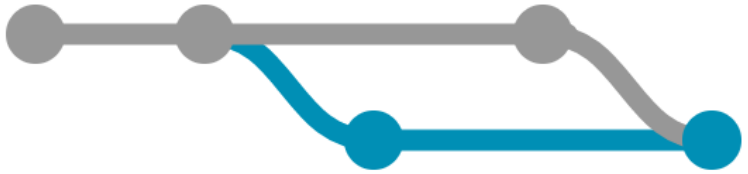
Branching Strategies for Every Occasion

One Size Fits All?

Specifically Tailored to Your Needs

Starting with Git: Branching 101

Git branches are different
Used for daily development



Instead of for “saving” milestones (SVN, TFS, …)

Let’s make the transition as easy as possible

Git Workflows

Centralized Workflow

Feature Branch Workflow

Gitflow Workflow

Forking Workflow

Dictator and Lieutenants Workflow

Guidelines

Centralized Workflow

Easiest way to get started

Do what you know

Treat as centralized repository

Developers pull and push directly

Just like SVN or TFS

No changes needed in how you work

Quicker adoption

Git = Advantages!

Mary

Ian

Centralized Workflow: Advantages with Git

Local & private repo

For each user

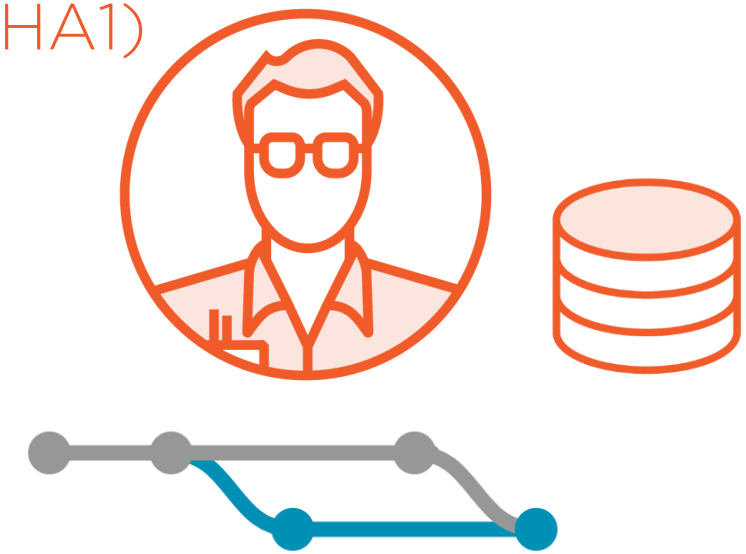
Git's branching model for development

Data corruption (SHA1)

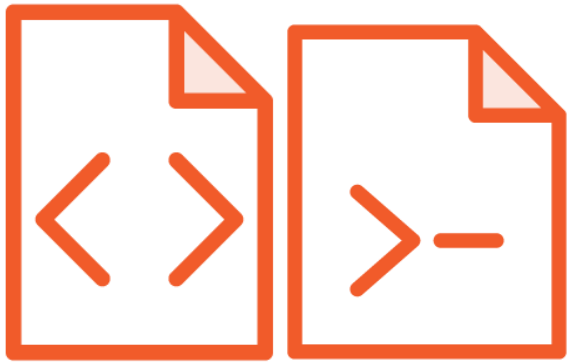
Faster

Full repo copy

Smaller

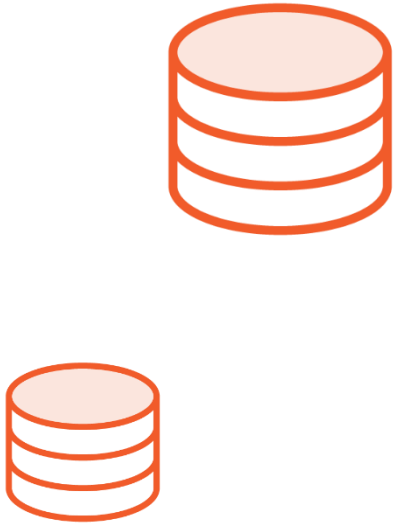


Centralized Workflow: Working



Work

Stage and commit locally



Push to origin

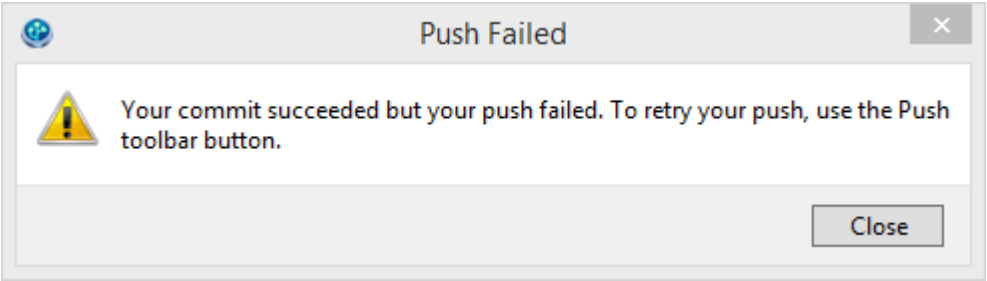
Centralized Workflow: Work Goes On

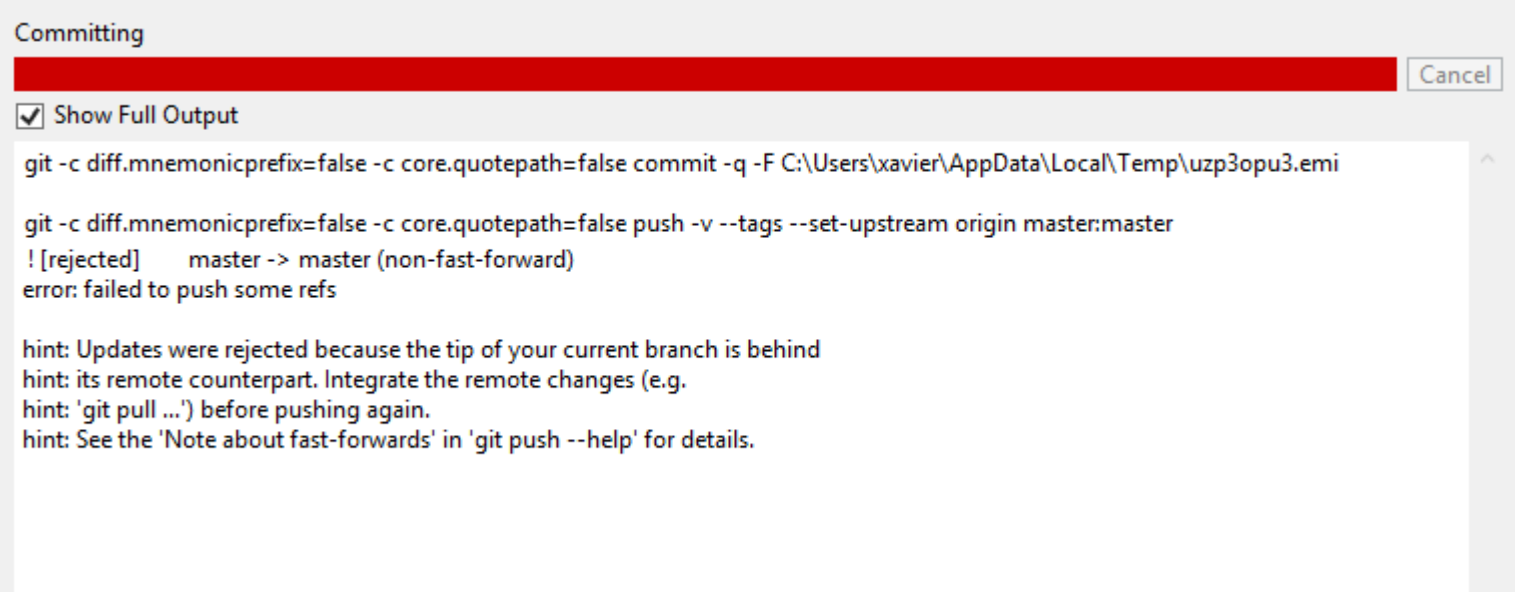
Developers keep working

Git will not allow overriding commits

Get all changes before Push

Resolve Conflicts and then Push





There is a secret to making it all work

Rebasing is the Key

Centralized Workflow: Rebasing

Add your changes in a linear way to origin/master
Focused changes
Clean history

Merge Commit | **Rebase**

Centralized Workflow

You Have Everything Under Control

But Git Can Do More

Problem with Centralized Workflow

Problem with Centralized Workflow

Code in *master* always deployment ready?

Work in progress on a feature?

Collaborating on a feature?

Prevent broken code in *master*?

Caveats present in other source control systems

Feature Branch Workflow

Feature Branch Workflow

Features
(or Bugs)

Dedicated Branch

Encapsulating
Work

Parallel Work

Collaboration
(Pull Requests)

Merge into Master

Integration: BitBucket, Jira & Bamboo

Git Repository

Issue Tracking &
Project Management

Continuous Integration
& Build Server

These are not the only options! Recommendation: Use the tools you know.

Feature Branch: A Branch for Every Feature

New work?

Checkout

Push, Pull Request and Merge

Work & Commit

Pushing Feature Branches

Merge	Pull Request
Incorporate changes	Request to merge
From current *feature branch*	Permissions
Into *master branch*	Notifications
Required: *Feature branch* up to date	Collaboration
Conflicts resolved	Code review (complete diff)
Merge commit	Discussion proposed feature
Or	Approved → Merged!
Fast-forward merge	[Capabilities vary Bitbucket vs GitHub vs ...]

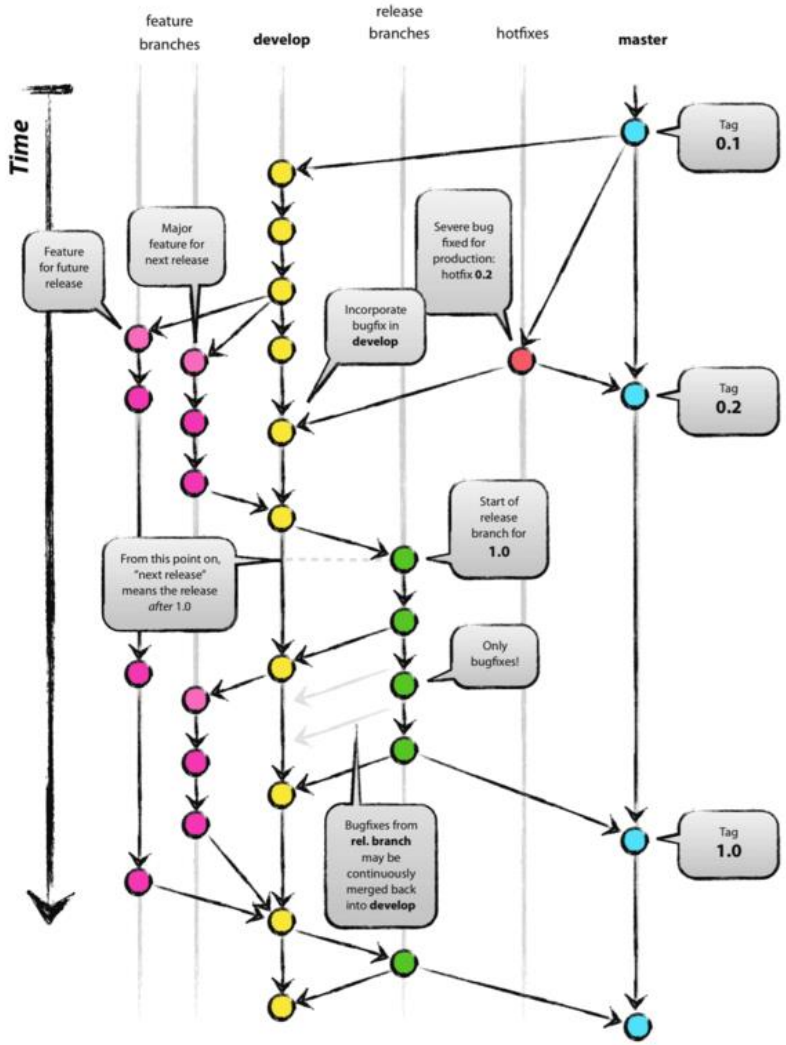
Gitflow Workflow

A successful Git branching model

By Vincent Driessen
on Tuesday, January 05, 2010

In this post I present the development model that I've introduced for some of my projects (both at work and private) about a year ago, and which has turned out to be very successful. I've been meaning to write about it for a while now, but I've never really found the time to do so thoroughly, until now. I won't talk about any of the projects' details, merely about the branching strategy and release management.

http://nvie.com/posts/a-successful-git-branching-model/



Gitflow: Brilliant!

That's what I think...

Gitflow Workflow

- Great complex projects
  - As well as small projects
- Brilliant organization of your work stream
- Simplifies complex source code management
  - Feature / Hotfix / Release
  - Parallel
- Distinction: Development vs. Production Ready
- How did I lived without it?

Gitflow Workflow: Main Branches

master

origin/master HEAD → Production Branch

Production Release:

Merge stable *develop* into *master*

via *Release Branch* and *Tag*

develop

origin/develop HEAD → Integration Branch

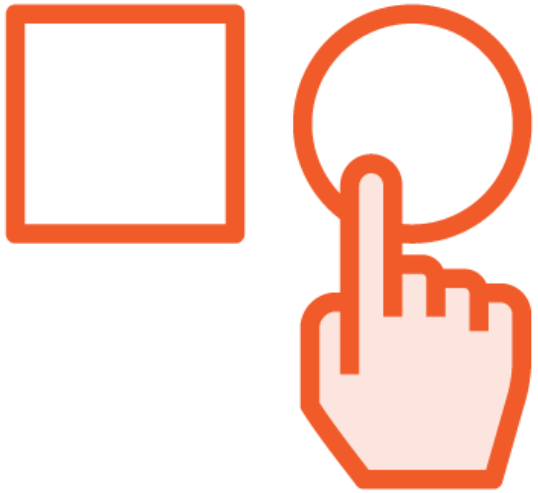
Latest dev changes

Nightly build

Only merge completed features

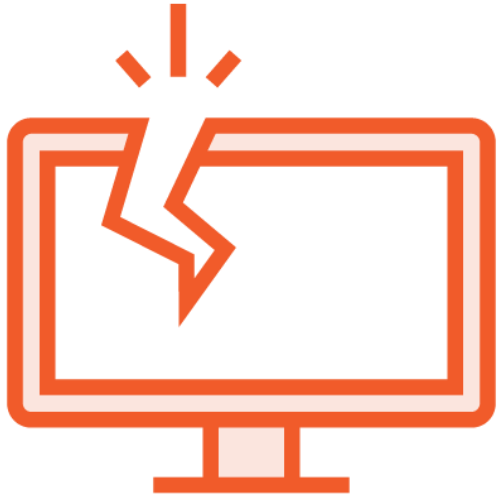
Infinite lifetime!

Gitflow Workflow: Supporting Branches



Feature

Release



Hotfix

Teams can work in parallel
Without much disruption

Feature Branches

A branch for every feature

Start by branching *develop*

Exists while in development

Recommended in local repo

origin: backup | collaboration

Ready?

Merge when done

a.k.a. “Topic” branches

Branch off from

develop

Merge back into

develop

Can be named anything except:

master, *develop*,

*release-**, *hotfix-**

Release Branches

Support preparation production release

- Branch almost production ready
- Can work on minor bug fixes
- Update metadata (version info)

Work can continue on *develop*

Ready?

- Merge into *master*
- *Tag* release and delete branch

May branch off from:

- *develop*

Must merge back into:

- *develop* and *master*

Branch naming convention:

- *release-**

Hotfix Branches

Prepare release (Unplanned! → React)

Fix a bug (or a few)

- While team keeps working

Ready?

- Update metadata
- Merge into *master* & *develop*
  - Except if *release* branch exists
  - Merge into *release* branch
- Delete branch

May branch from

- *master*

Must merge back into

- *develop* and *master*

Potentially *release*

Naming convention

- *hotfix-**

Using GitFlow from SourceTree

SourceTree

File Edit View Repository Actions Tools Help

Clone / New | Commit | Checkout | Discard | Stash | Add | Remove | Add/Remove | Fetch | Pull | Push | Branch | Merge | Tag | Git Flow | Terminal | Settings

Pluralsight
Usados.cr
using-git-with-gui D:\Drop
1 1 | master
gitgraph D:\Dropbox\xavie
| develop
solrnet D:\github\solrnet
| develop

gitgraph | using-git-with-gui | solrnet

File Status
Working Copy
Branches
develop
master
Tags
Remotes
origin

All Branches | Show Remote Branches | Date Order | Jump to:

Graph | Date | Author | Commit

Update Hi | May 2015 11:11 | xaviermorera <xavi | cc6ae59
origin | May 2015 11:07 | xaviermorera <xavi | df9fdcc
origin | May 2015 11:06 | xaviermorera <xavi | 53ea81e
origin | May 2015 11:04 | xaviermorera <xavi | 6d4243f
0.5.0a | May 2015 6:09 | mausch <mauricic | 0c425c0
Nuget pac | May 2015 18:16 | mausch <mauricic | 226483d
Bump vers | May 2015 17:20 | mausch <mauricic | 2694a37
Update ms | May 2015 17:07 | mausch <mauricic | 3ce18aa
Update RE | May 2015 10:51 | Mauricio Scheffer | f005547
0.4.0 | May 2015 10:21 | mausch <mauricic | eb2f2f3
Bump vers | May 2015 10:17 | mausch <mauricic | acf277c
Upgrade project to VS2013 | 3 May 2015 9:33 | mausch <mauricic | 17bf323
Update FAQ.md | 2 May 2015 13:19 | Mauricio Scheffer | 1c716e8

Start New Feature

Feature Name:

Start at: Latest development branch
Working copy parent
Specified commit: ...

Preview
feature/ Create new branch
develop Latest development branch
...

OK | Cancel

Sorted by path | Search

Commit: df9fdcc93d11aa2bf9296b0cc53a33469d1f5d2b [df9fdcc]
Parents: 77f56afb3b
Author: xaviermorera <xavier@familiamorera.com>
Date: Tuesday, May 5, 2015 11:07:26 AM
Labels: HEAD, origin/master, origin/HEAD, master, develop

Update Highlighting

Documentation/Highlighting.md

Reverse hunk

```
# Highlighting
```

Demo

Gitflow Workflow

From SourceTree

Forking Workflow

Important Difference

Besides main repository

Every developer (or team)

- Server side repository (usually public)
- Local (private)

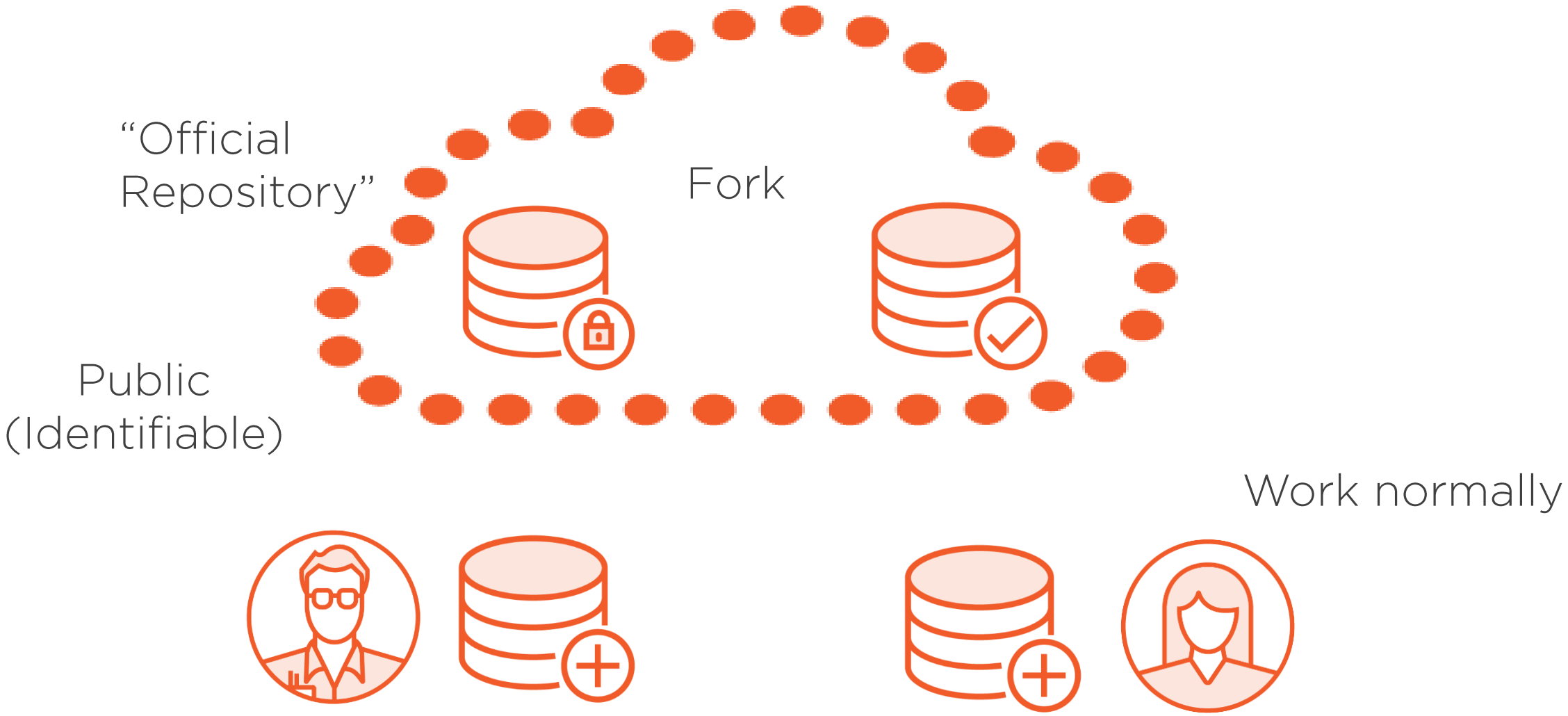
Push to their own server side repo

Pull requests to main repository

Perfect for open source!

As well as enterprise

Forking Workflow



Forking from Repository

GitHub | This repository | Search | Explore | Features | Enterprise | Pricing | Sign up | Sign in

mausch / SolrNet

Watch 97 | Star 479 | Fork 390

Code | Issues 60 | Pull requests 19 | Pulse | Graphs

Bitbucket | Dashboard | Teams | Repositories | Snippets | Create

pluralsight-demo
Forking demo

ACTIONS

- Clone
- Create branch
- Create pull request
- Compare
- Fork

Repository setup

Your repository is empty — let's put some bits in your bucket.

Get code into Bitbucket fast using Atlassian SourceTree or the command line

SourceTree

Get started using the SourceTree client

Repository Might Keep Evolving

Synchronize

Forking Workflow

Pull request

Dictator and Lieutenants Workflow

- Linux uses this workflow
- Benevolent dictator: Linus Torvalds
- Blessed repository
    - Developers pull from here
- Developers
    - Rebase on master
- Lieutenants merge into their branch
- Dictator merges Lieutenants' branches
- Dictator pushes to blessed repository

Takeaway

Centralized Workflow	Feature Branch Workflow	Gitflow Workflow
Forking Workflow	Dictator and Lieutenants Workflow	Pick the One That Works for You