

# Transformative Git Practices

New Techniques for Hyper-productivity

Nicola Paolucci



Platinum Sponsors:



#DevoxxPL

@durdn

Special Night & Time

#GOLD RUSH

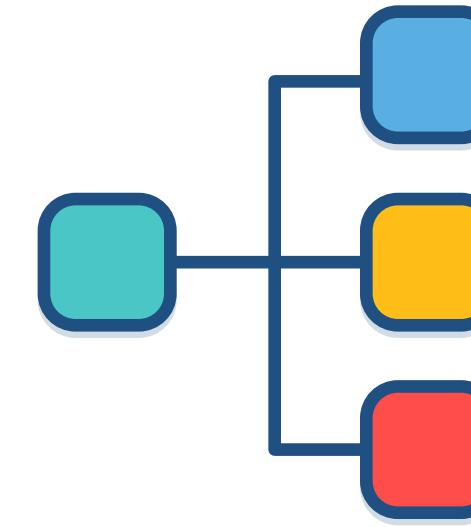
MONDAY  
8/7c

git push --force

KLONDIKE

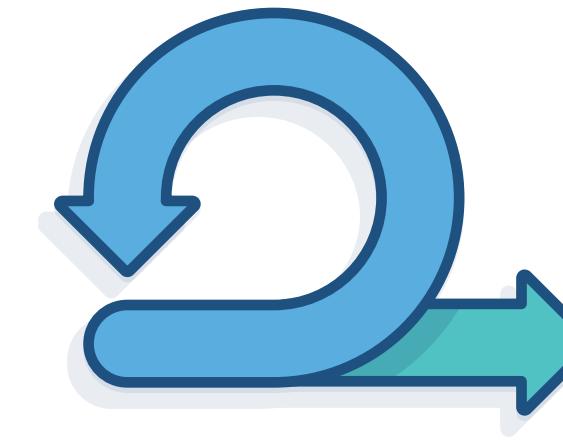
In 2 Days

# Today we'll cover some powerful Git goodies



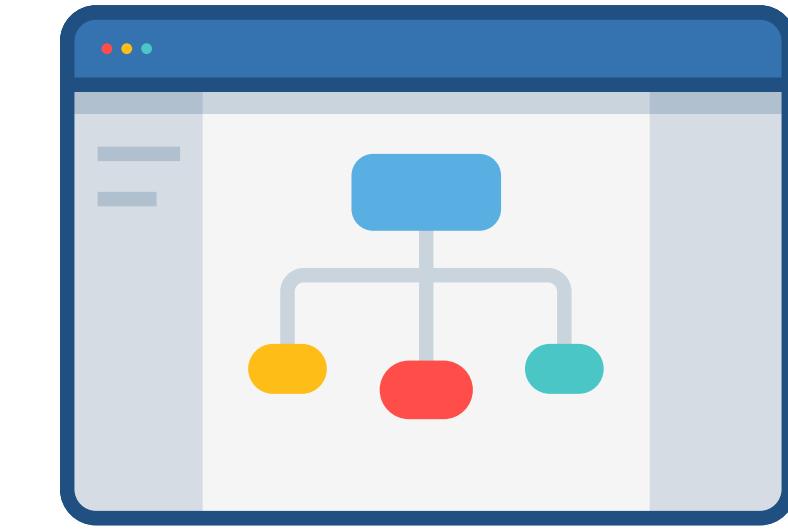
## Worktree clones

For when you have to work concurrently on multiple long running branches



## The magic of --fixup

A powerful technique to cleanup your history before sharing with it the team



## Painless sub-projects

You can use git subtree to handle external libraries in a clean and efficient way

and a final, cool topic...

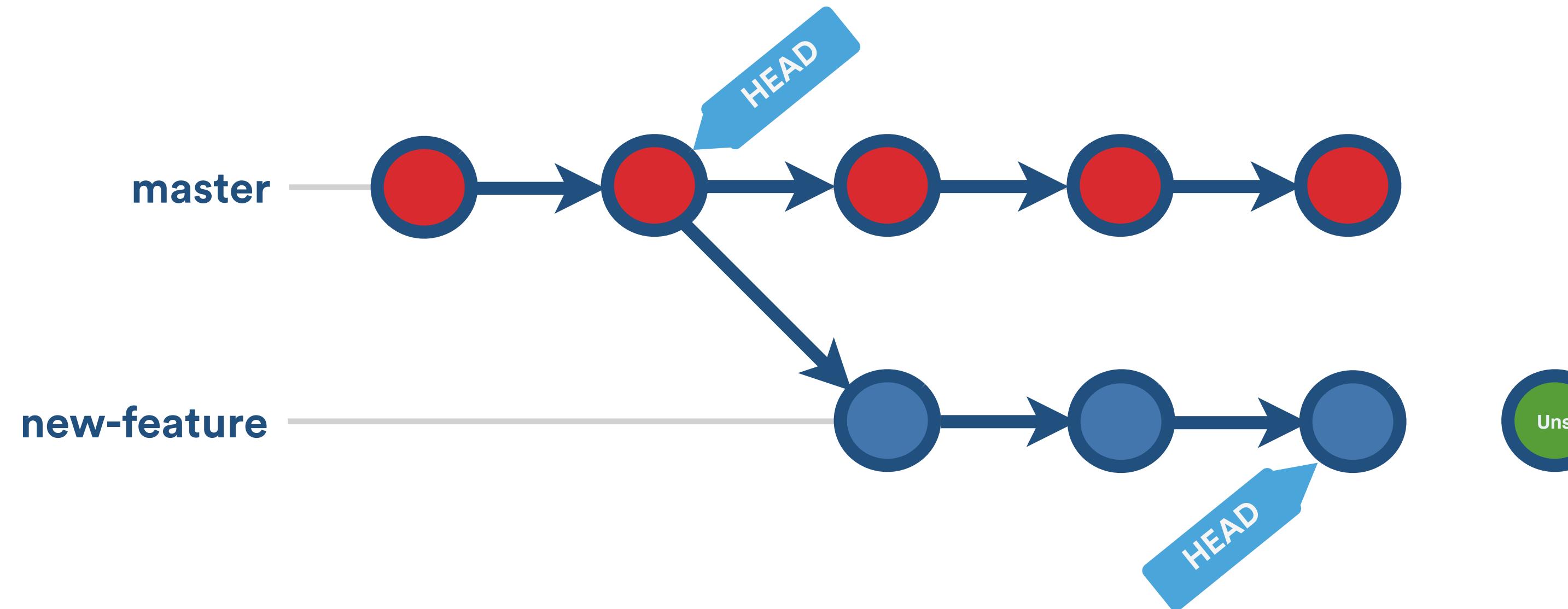


# Git Worktree

Local clones for parallel work

# The old simple way: branch, commit, push

```
git checkout -b new-feature
```



It's clumsy to work in parallel  
on multiple branches

# To work on 2+ branches at the same time



It leaves hanging  
uncommitted work  
lying around



You either work  
around it having  
intermediate WIP  
commits



Or are a pro at using  
the Stash (which is  
cool btw)



# git stash is awesome

It's a way to temporarily store your unsaved work  
on a stack of patches



# But there's a new way: git worktree

Available since Git 2.5

# git worktree creates additional clones inside your project

```
git worktree add .  
-b feature/PRJ-1-new-fancy-button  
fancy-button
```

..... Add a new worktree

..... Branch to checkout in clone

..... Folder where to checkout



# git worktree list - shows the worktrees created

## It shows the folder

Where the additional cloned has been checked out

## And the branch is pointing at

So that you have an immediate overview of all the trees you have created

```
$ git worktree list
```

```
/Users/np/p/dac
```

```
/Users/np/p/dac/git-the-all-new-modern-workflow
```

```
/Users/np/p/dac/provision-a-cluster-with-rancher
```

```
/Users/np/p/dac/static-site-with-pipelines-and-middleman
```

```
f1f7f8b [master]
```

```
7fe9096 [blog/git-the-all-new-modern-workflow]
```

```
b4b84fb [blog/provision-a-cluster-with-rancher]
```

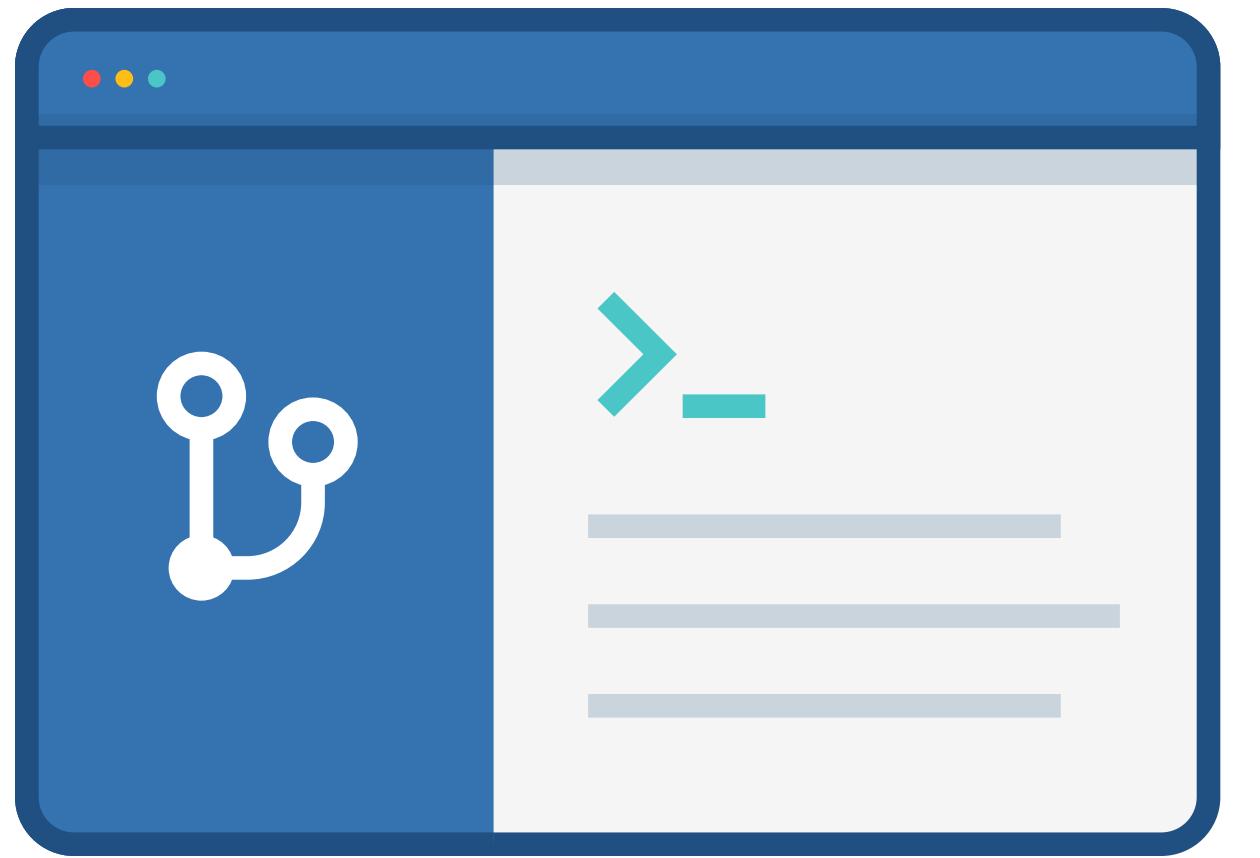
```
d6bd9fd [blog/static-site-with-pipelines-and-middleman]
```



# git worktree prune - once you're done with a clone

```
$ rm -rf fancy-button  
$ git worktree prune
```



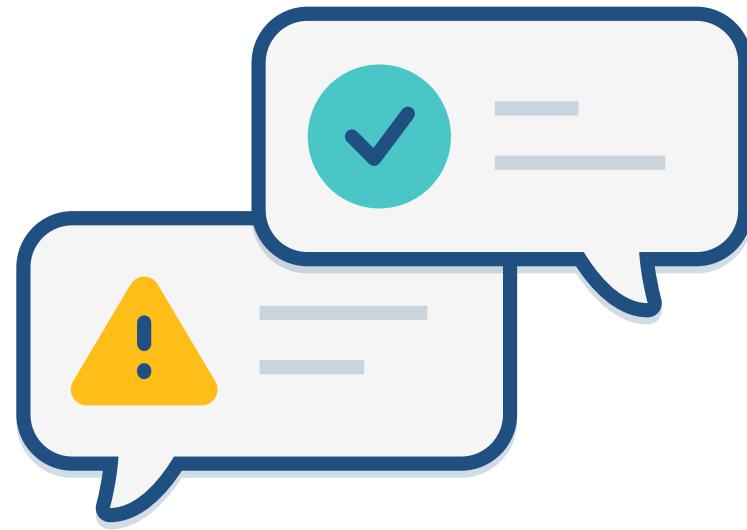


## Demo Time!

Let's see how “git worktree” works in practice



# Why git worktree is a great choice



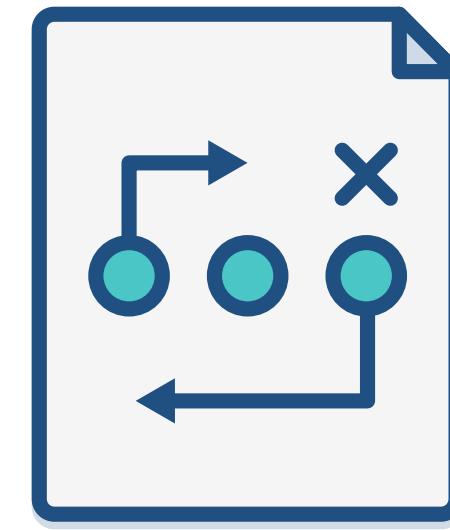
## Better work in parallel

Having multiple clones with all the branches you're working on available in your repo is a big win.



## Keep working while testing

If you have a big battery of tests which are slow to run, you can run them in a worktree clone while you keep working.



## Replace the normal branching?

When you get used to worktrees, they become very natural for any feature branch work.





# The magic of --fixup

Smooth history cleanup for perfectionists



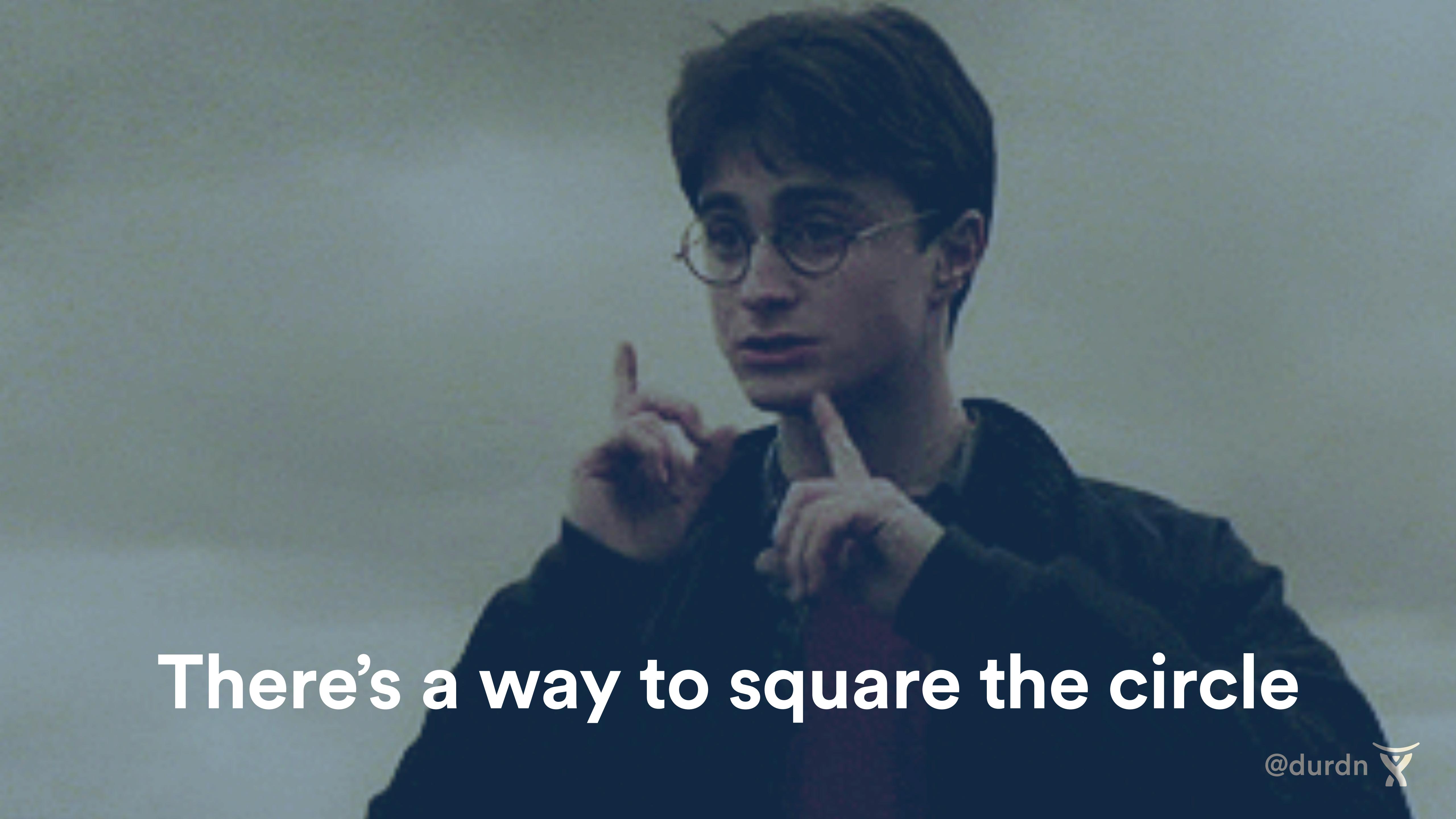
It's great to work in small chunks

**There's a tension between...**

Commits as  
throw away  
save points

Commits as a  
polished  
meaningful way  
to communicate  
a feature

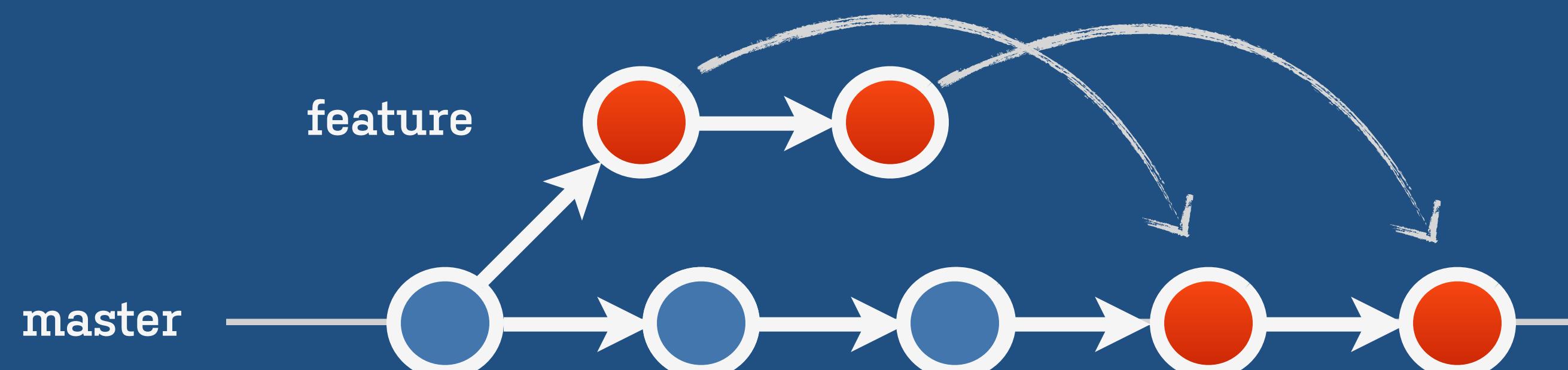


A black and white photograph of a man with dark hair and glasses, wearing a dark turtleneck sweater. He is looking slightly to the left of the camera with a thoughtful expression, his hands raised in front of him with fingers interlocked. The background is a plain, light-colored wall.

**There's a way to square the circle**

# What is an --interactive rebase?

Helps you clean up your private branches  
before publishing them interactively



# First step: Turn on the “autosquash” feature

```
$ git config --global rebase.autosquash true
```



Work in small throwaway bits, use  
commits as save points  
as you please

# Annotate your raw commits with “--fixup” or “--squash”

## Fixup

Use fixup to quickly fix problems in a commit, without the need to change or add to the commit message

## Squash

For additions to a commit where you want to add also an additional note to the commit message

```
$ git commit --fixup <commit>
```

```
$ git commit --squash -m"Add to feature" <commit>
```



# Annotate your raw commits with “--fixup” or “--squash”

## Fixup

Use fixup to quickly fix problems in a commit, without the need to change or add to the commit message

## Squash

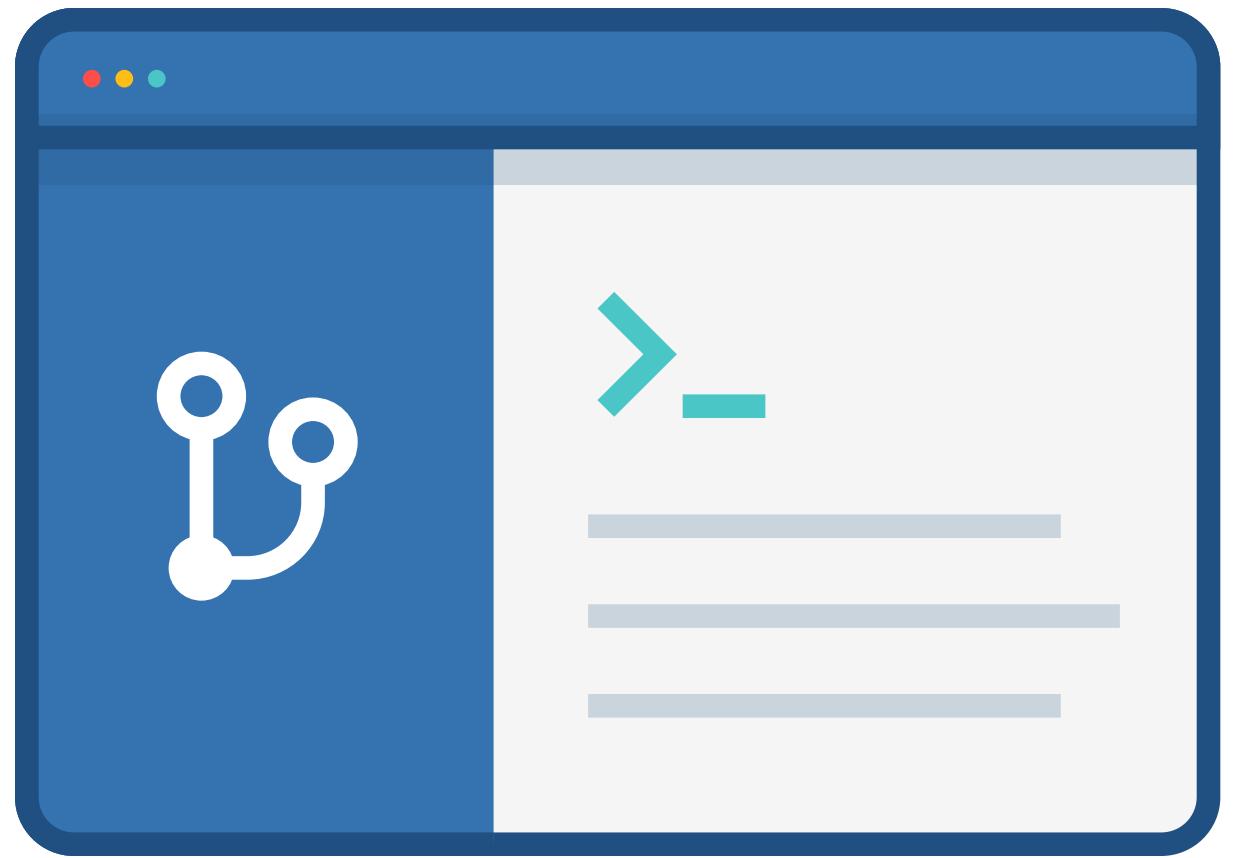
For additions to a commit where you want to add also an additional note to the commit message

```
$ git commit --fixup :<string pattern>
```

```
$ git commit --fixup :/PRJ-123
```



# Rebase is now pre-filled and easy



## Demo Time!

Let's see how “--fixup” and “--squash” work  
in practice



A photograph of two people, a man and a woman, looking down at a screen. The screen displays a pattern of colored dots (blue, green, red, yellow) on a dark background. The man is on the left, wearing a dark turtleneck sweater. The woman is on the right, wearing a light-colored top. They appear to be engaged in a task or discussion related to the content on the screen.

Let's talk about project dependencies

# git subtree

## Introduced in 1.7.11

Alternative to git submodule to handle external dependencies.

## Inject dependency

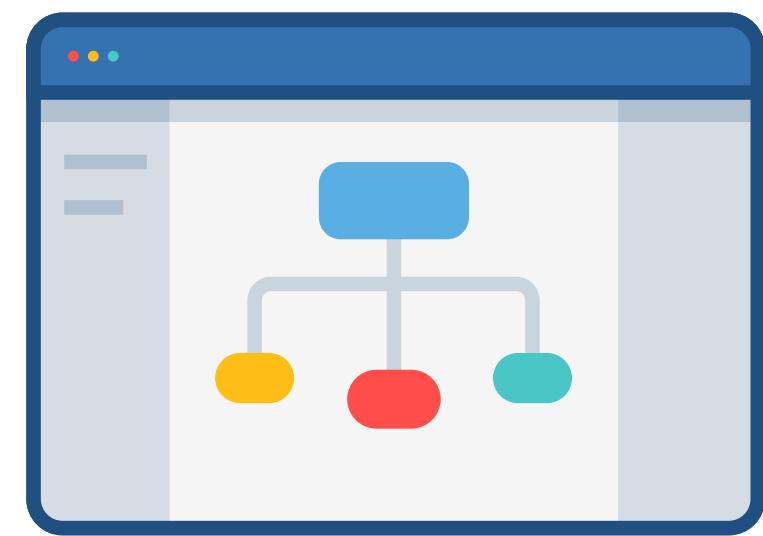
It allows you to inject an external project into a sub-folder

## Extract project

It can be also used to extract a sub-folder as separate project

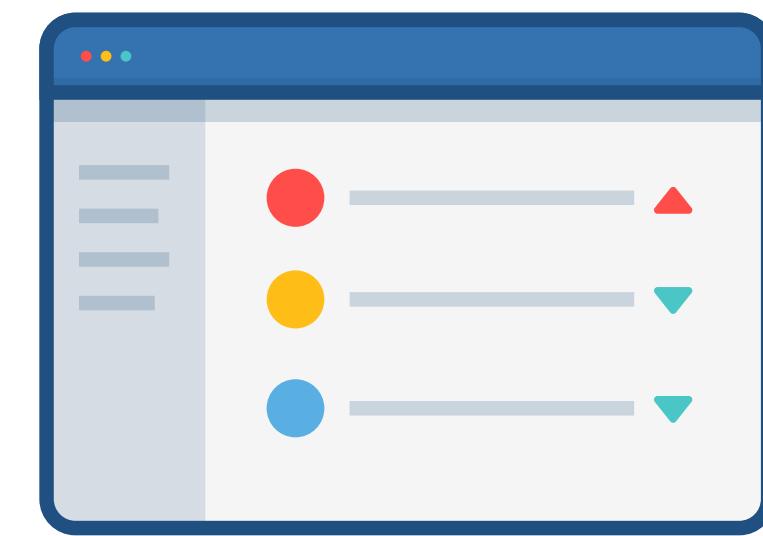


# When and why is git subtree a great choice



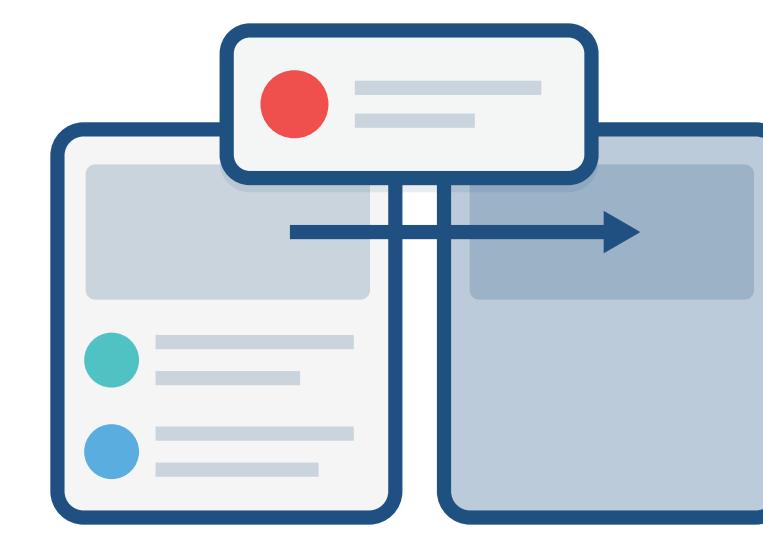
## No training

Does not require your entire team to be trained in the use of the command



## Stores in regular commits

The command stores the external dependency in regular Git commits. Squashing the history optionally.



## Clean integration points

It makes it easy to see when integrations happened and makes it easy to revert them.



# Syntax to inject a project

```
git subtree add  
--prefix target-folder  
https://bitbucket.org/team/sub.git \\\n  v1.1 --squash
```

..... **Command to inject project**

..... **Folder where to insert code**

.... **Repository URL**

# Under the hood of git subtree

commit ab54c4e0b75c3107e3e773ab9b39268abddca002

Author: Nicola Paolucci <npaolucci@atlassian.com>

Date: Tue Sep 29 15:27:35 2015 +0200

Squashed ‘src/sub-project’ content from commit df563ed

git-subtree-dir: src/sub-project

git-subtree-split: df563ed15fa6...6b2e95d3

# Result of git subtree add

```
commit ab54c4e0b75c3107e3e773ab9b39268abddca002
```

```
Author: Nicola Paolucci <npaolucci@atlassian.com>
```

```
Date: Tue Sep 29 15:27:35 2015 +0200
```

```
Squashed '.vim/bundle/fireplace/' content from commit df563ed
```

```
git-subtree-dir: .vim/bundle/fireplace
```

```
git-subtree-split: df563ed15fa685ce2508bf16b3ca7e176b2e95d3
```

```
commit 8fb507baf7b270c30c822b27e262d0b44819b4c5
```

```
Merge: 606cd3e ab54c4e
```

```
Author: Nicola Paolucci <npaolucci@atlassian.com>
```

```
Date: Tue Sep 29 15:27:35 2015 +0200
```

```
Merge commit 'ab54c4e0b75c3107e3e773ab9b39268abddca002' as '.vim/bundle/fireplace'
```

# To keep the sub-project up to date

```
git subtree pull  
--prefix target-folder  
https://bitbucket.org/team/sub.git \  
v1.5 --squash
```

..... **Command to pull project**

..... **Folder where to insert code**

.... **Repository URL**

# Under the hood of git subtree

commit ab54c4e0b75c3107e3e773ab9b39268abddca002

Author: Nicola Paolucci <npaolucci@atlassian.com>

Date: Tue Sep 29 15:27:35 2015 +0200

Squashed ‘src/sub-project’ content from commit df563ed

git-subtree-dir: src/sub-project

git-subtree-split: df563ed15fa6...6b2e95d3

# Find the symbolic ref matching a hash (sha-1)

```
git ls-remote https://bitbucket.org/team/sub.git | grep <sha-1>
```

```
git ls-remote https://bitbucket.org/team/sub.git | grep df563ed  
df563ed15fa685ce2508bf16b3ca7e176b2e95d3 v1.1  
5eaff1232acedeca565er7e1333234dacccebfff v1.5
```

..... **Git plumbing to list all remote refs**

..... **sha-1 of last commit pulled**

..... **Repository URL**

# Aliases to make your life easier!

..... Alias section of your .gitconfig

<http://bit.do/git-aliases>

```
[alias]
```

```
sba = "!f() { git subtree add --prefix $2 $1 master --squash; }; f"
sbu = "!f() { git subtree pull --prefix $2 $1 master --squash; }; f"
```

# How to use the aliases

```
git sba <repo URL> <destination-folder>
```

```
git sba https://bitbucket.org/team/sub.git src/sub
```

# When NOT to use git subtree



**When you have many dependencies**



**When you have constant updates to your dependencies**



**When everyone in the team must work on sub-projects**





For complex project dependencies  
Use a dependency tool.

Really.

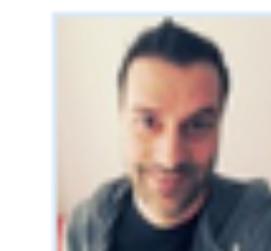
# Alternatives? Read my rant on project dependencies

<http://bit.do/git-dep>



All Blogs   Products   Developer   News

## Git and project dependencies



By [Nicola Paolucci](#)

Developer

On April 16, 2014

Part of our

**Git  
Together**

series on team collaboration v

Consider the following questions:

How do you handle project dependencies with `git` ?



# How to extract a project

Let's learn how to use subtree **split**

# Git subtree to extract a project

```
git subtree split  
--prefix my/project/  
--branch extracted
```

..... **Command to split out project**

..... **Folder prefix to extract**

..... **where we store it**

```
git rm -rf my/project
```

## Remove from old repo

We can remove the contents of the folder from the repo

```
git init  
git pull ../path/ extracted
```

## Import extracted branch

Initialise a new repo and import the extracted branch

```
git remote add origin ...  
git push origin -u master
```

## Push to new remote

After we imported the code we can push it to a new repository





# Have a Pipeline with your repo

Build your project in the cloud



orochi

Atlassian / Project: Atlassian / orochi

## Commits

All branches ▾

Find commits

ACTIONS	Author	Commit	Message	Date	Builds
Clone	Mark Adams	<a href="#">8b91a46 M</a>	Merged in madams/update-pipelines (pull request #64) Update Pipelines YML to v2 format	5 days ago	
Create branch	Mark Adams	<a href="#">7205d72</a>	Update Pipelines YML to v2 format	5 days ago	
Create pull request	Daniil Penkin	<a href="#">10dc8e9 M</a>	Merged in dpenkin/BB-15986-large-file-diff-support (pull request #63) BB-15986: Support diff in...	2016-04-22	
Compare	Daniil Penkin	<a href="#">71c267f M</a>	Merged in dpenkin/BB-15986-large-file-pointer-support (pull request #62) BB-15986: Implemen...	2016-04-22	
Fork	Daniil Penkin	<a href="#">ff80c64</a>	BB-15986: Apply last changes to basic LFS support in Orochi	2016-04-21	
Overview	Daniil Penkin	<a href="#">98cd315</a>	BB-15986: PR review - Add caching for large file info calculation	2016-04-21	
Source	Daniil Penkin	<a href="#">db345a7</a>	BB-15986: Implement large file diff info for git	2016-04-21	
Commits	Daniil Penkin	<a href="#">fdb3060</a>	BB-15986: PR review - minor refactoring - Make largefileinfo a method instead of a property - R...	2016-04-21	
Branches	Daniil Penkin	<a href="#">e153516</a>	BB-15986: PR review - Reformat code, rename attributes	2016-04-15	
Pull requests <span>6</span>	Daniil Penkin	<a href="#">36262fa</a>	BB-15986: PR review - minor fixes and changes - Fix lower LFS pointer size bound, adjust test...	2016-04-15	
Downloads	Daniil Penkin	<a href="#">b70b52b</a>	BB-15986: Implement large file info for GitFile	2016-04-14	
Builds	Brodie Rao	<a href="#">673e1bd</a>	git: upgrade to git 2.7.4.1.g5468f9e (HOT-38581) This upgrades Orochi and its tests to use Git ...	2016-04-07	
	Mark Adams	<a href="#">86b5057 M</a>	Merged in madams/pipelines (pull request #61) Added support for Bitbucket Pipelines builds	2016-04-06	
	Mark Adams	<a href="#">a6aac3a</a>	Added support for Bitbucket Pipelines builds	2016-04-06	
	Jonathan Robson	<a href="#">8b33b4a M</a>	Merged in jrobson/BB-15349-rename-threshold (pull request #53) [BB-15349] Support configur...	2016-04-05	
	Jonathan Robson	<a href="#">d403864</a>	Make 50 the minimum rename threshold	2016-04-05	
	Jonathan Robson	<a href="#">40fd7ea</a>	Consolidate rename arguments	2016-04-04	
	Sean Farley	<a href="#">6bd84f2</a>	hg: fix apparent misspelling of precommit -> pretxncommit	2016-03-29	

Bitbucket Teams Projects Repositories Snippets Feedback Find a repository... ?

sass-hugo

ACTIONS

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

NAVIGATION

- Overview
- Source
- Commits
- Branches
- Pull requests
- Downloads
- Pipelines
- Settings

Nicola Paolucci [AMS] / sass-hugo

## Settings

GENERAL

- Repository details
- Access management
- Branch management
- Username aliases
- Deployment keys
- Compliance
- Transfer repository
- Delete repository

INTEGRATIONS

- Services
- Webhooks
- Links

PULL REQUESTS

- Default reviewers
- Merge restrictions

ISSUES

- Issue tracker settings

WIKI

- Wiki settings

Pipelines BETA

Enable Pipelines, copy the sample configuration, and add the bitbucket-pipelines.yml file to start building on every commit.

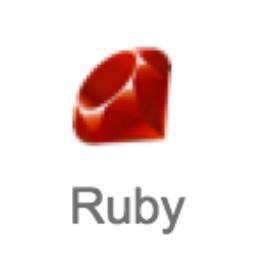
1 Enable pipelines  X

Bitbucket will now build on every push once you have a **bitbucket-pipelines.yml** configuration file in the root directory.

2 Copy the sample configuration

Copy the example and paste it in the file that we'll open in the next step.

Currently we don't support multiple steps. The example below is a simple step where *script* is a sequence of commands that run each build.

 Node.js  Java  Ruby  Python  PHP  Advanced

Learn more about [how to configure bitbucket-pipelines.yml](#) or [discover our integrations](#).

```
1 # You can use a Docker image from Docker Hub or your own container
2 # registry for your build environment.
3 image: node:5.11.0
4
5 pipelines:
6   default:
7     - step:
8       script: # Modify the commands below to build your repository.
```

 orochi

**ACTIONS**

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

---

**NAVIGATION**

-  Overview
-  Source
-  Commits
-  Branches
-  Pull requests 6
-  Downloads
-  Builds

---

[Send feedback](#)

atlassian / orochi

## Pipelines BETA

Successful  

⌚ Duration: 3 min 47 sec  
Started: 5 days ago  
[View configuration](#)

⌚ 7205d72  
Commit message unavailable  


 madams/update-pipelines

## Logs

```
> docker pull atlassian/agent-setup-git:latest
> docker run -t --entrypoint=/bin/bash --memory=2048m -v /tmp/bashScript4695982058...
> git clone --branch='madams/update-pipelines' --depth 50 https://x-token-auth:$AU...
> docker pull centos:7
> docker run -t --rm=true --cidfile=/tmp/1461852093387-0/{6c07d754-37b4-4504-8a9a...
> scripts/install_linux_dependencies.sh
> pip install -r requirements.txt
```

# Deploy to any of these and more!



# Bitbucket Pipelines BETA

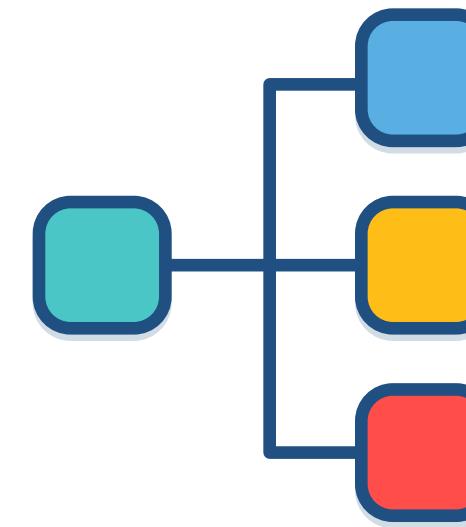
Build, test and deploy from Bitbucket

[Sign up for beta](#)

---

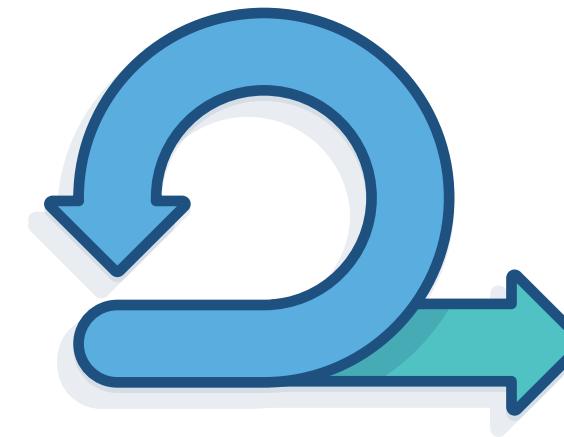
<http://bit.do/pipelines>

# We've covered some powerful Git goodies



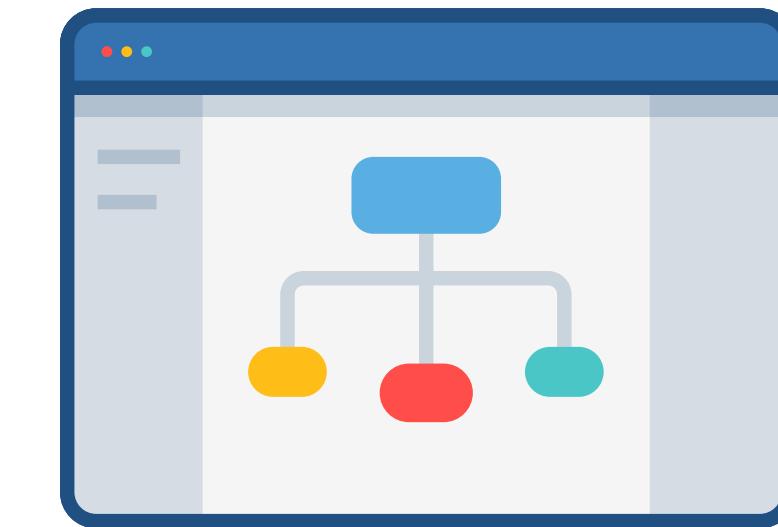
## Worktree clones

For when you have to work concurrently on multiple long running branches



## The magic of --fixup

A powerful technique to cleanup your history before sharing with it the team



## Painless sub-projects

You can use git subtree to handle external libraries in a clean and efficient way





Thank you! Follow me @durdn



# Thank you! Follow me @durdn

[Features](#)[Integrations](#)[Server](#)[Pricing](#)

# <http://bit.do/pipelines>

## Bitbucket Pipelines BETA

Build, test and deploy from Bitbucket

[Sign up for beta](#)