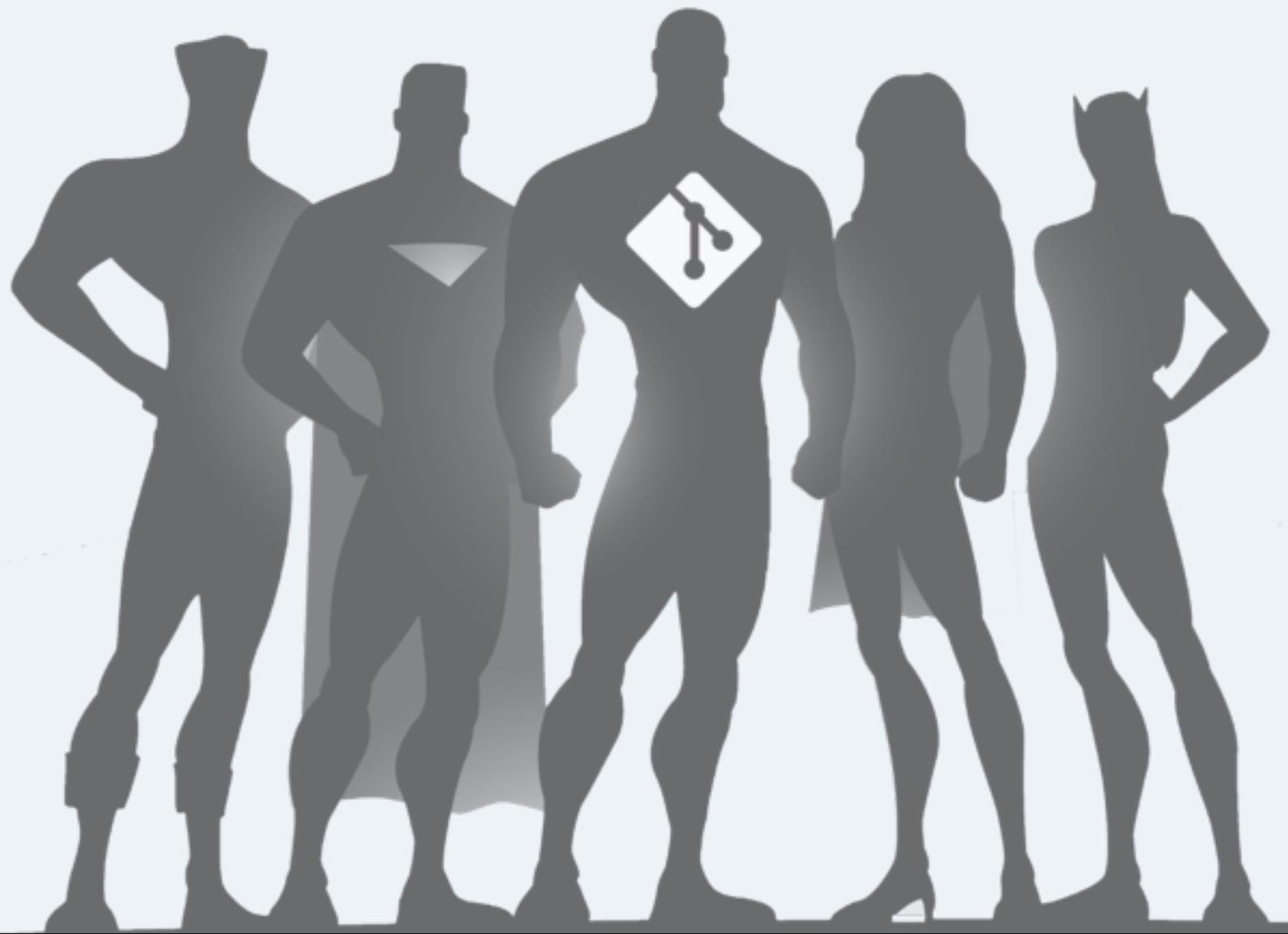


GIT

WITH T FOR

TEAM

FREE BEER



@svenpet

SWN

GIT

MERCURIAL

CIVS

ANYTHING ELSE

TFS ★ *Perforce* ★ *ClearCase* ★ *VSS*

Movies



you must love pain

WHY IS **GIT** GOOD*?

**CHEAP BRANCHING
EVERYTHING IS LOCAL**

GIT IS FAST

GIT IS SMALL

THE STAGING AREA

DISTRIBUTED

GITHUB & BITBUCKET

GIT IS THE NEW STANDARD

HUGE COMMUNITY

* from Lemi Orhan Ergin

WHY IS **GIT** GOOD*?

**CHEAP BRANCHING
EVERYTHING IS LOCAL**

GIT IS FAST

GIT IS SMALL

THE STAGING AREA

DISTRIBUTED

WORKFLOW CAPABILITIES

GIT IS THE NEW STANDARD

HUGE COMMUNITY

* from Lemi Orhan Ergin



MASTER

RELEASE

VERSION 1.9

DETAILED VIEW



BIRDS VIEW



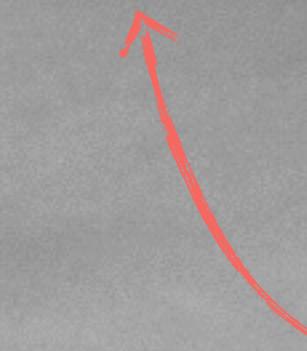
git

DESIGNED FOR OPEN SOURCE

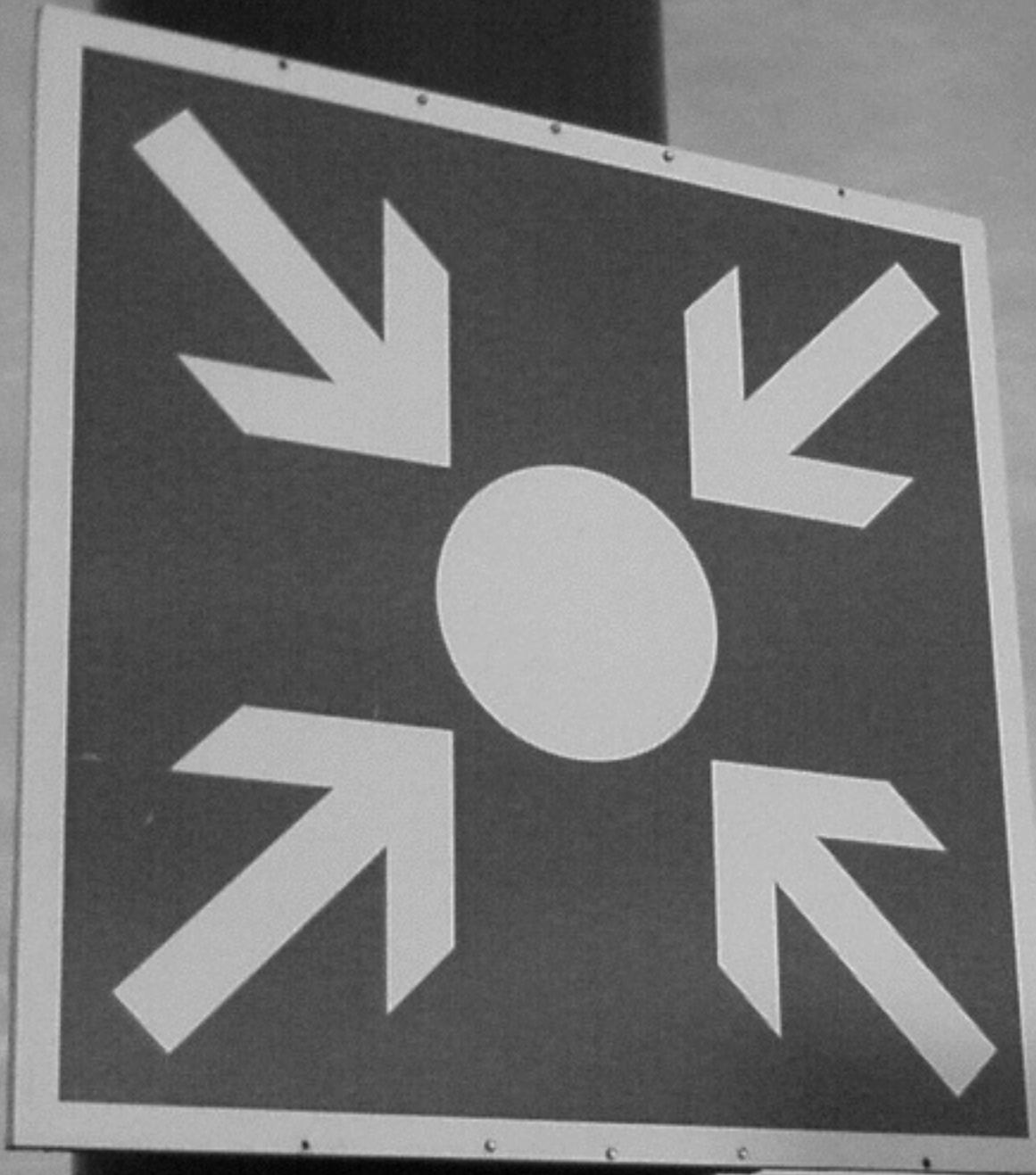


DISTRIBUTED WORKFLOW

CENTRALIZED WORKFLOW



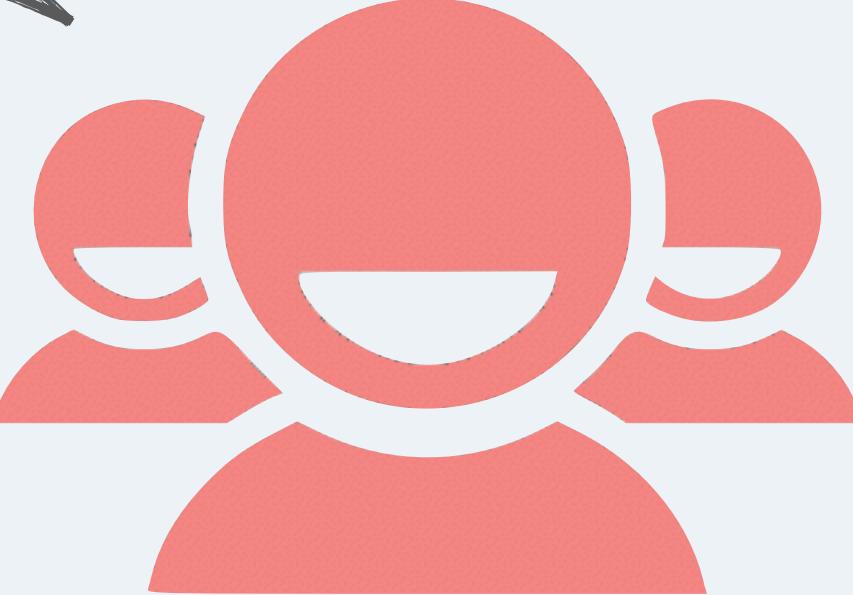
companies like those





Sven Peters
ATLASSIAN AMBASSADOR

@svenpet
<http://svenpet.com>



Can we fix a bug for the upcoming
RELEASE ?

Can we do a fast
HOTFIX for the current version?

Is the code for that
FEATURE complete?

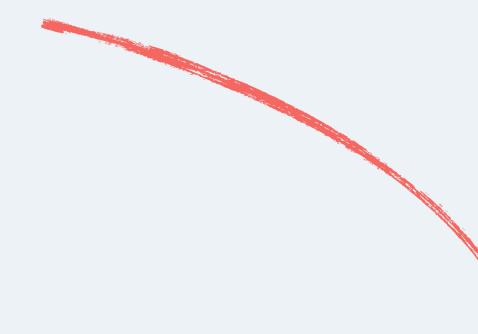
Has everybody
REVIEWED the code for this feature ?

I FIND YOUR LACK OF **WORKFLOW STRUCTURE** DISTURBING



DIFFERENT
PEOPLE
PRODUCTS
CULTURES

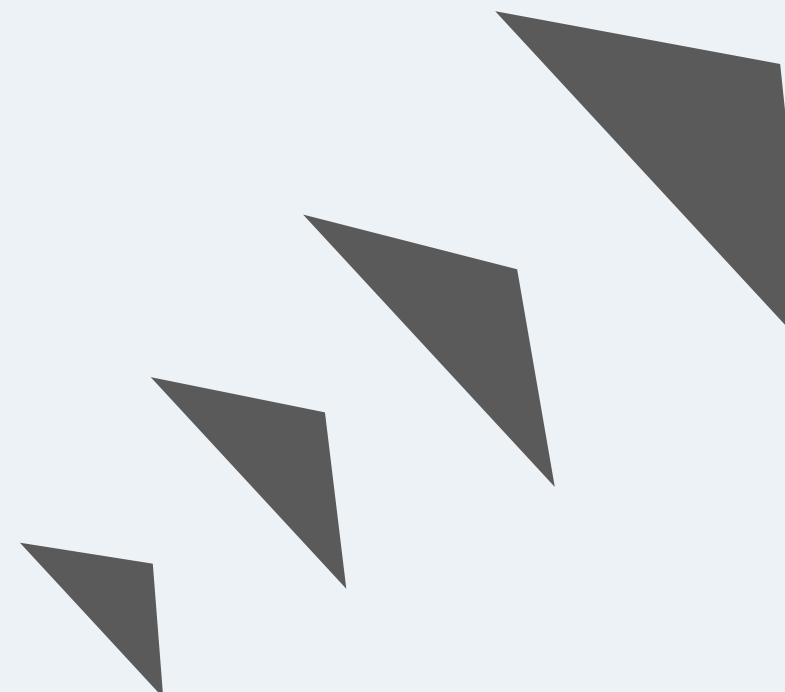
**DIFFERENT
PEOPLE
PRODUCTS
CULTURES**



JIRA, Bitbucket, Stash

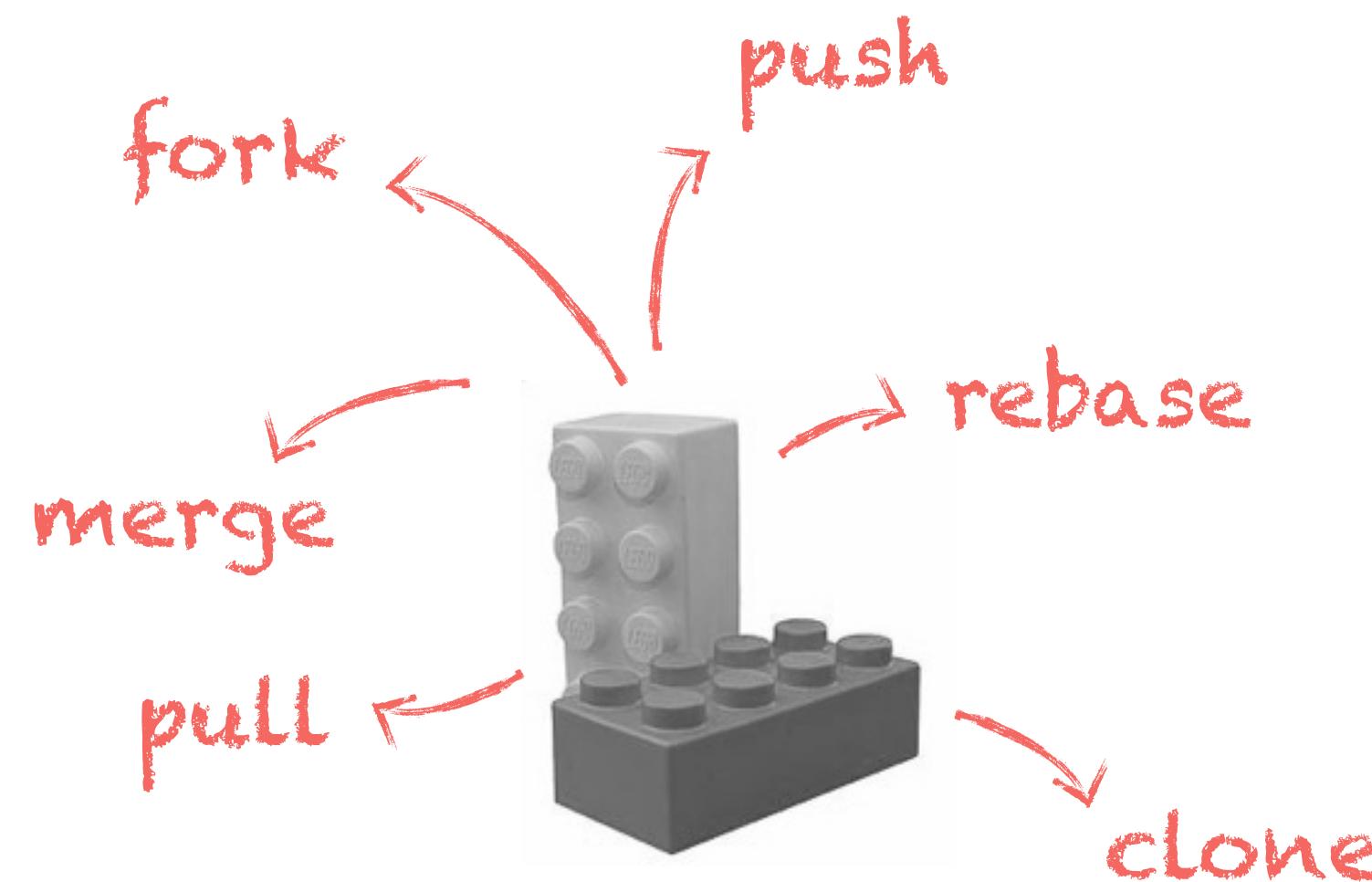
DIFFERENT
PEOPLE
PRODUCTS
CULTURES

DIFFERENT
PEOPLE
PRODUCTS
CULTURES



DIFFERENT
WORKFLOWS

BUILD YOUR WORKFLOW



BUILD YOUR **WORKFLOW**



**ONE WORKFLOW:
NOT POSSIBLE WITH GIT**

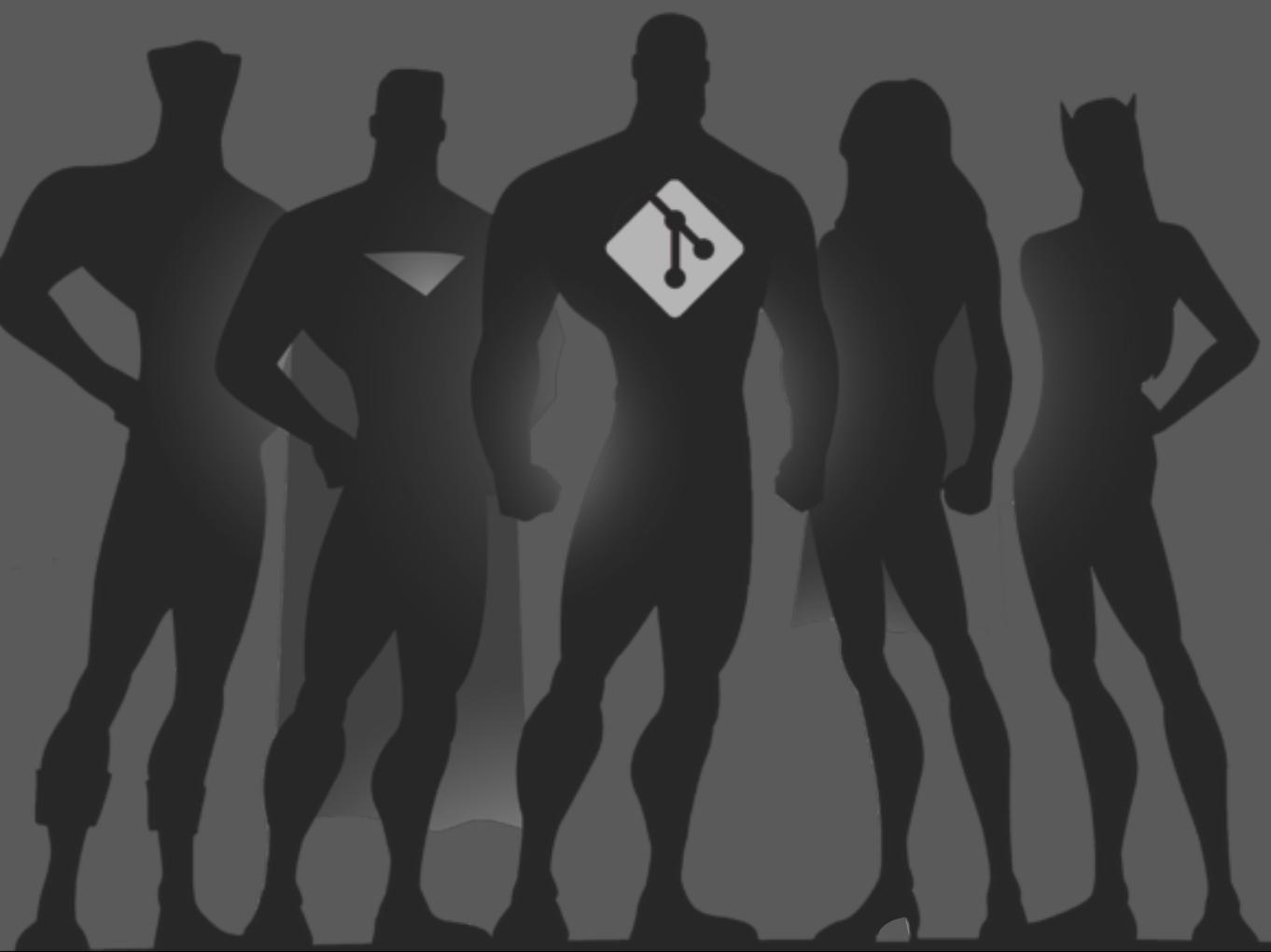
block files while working on it

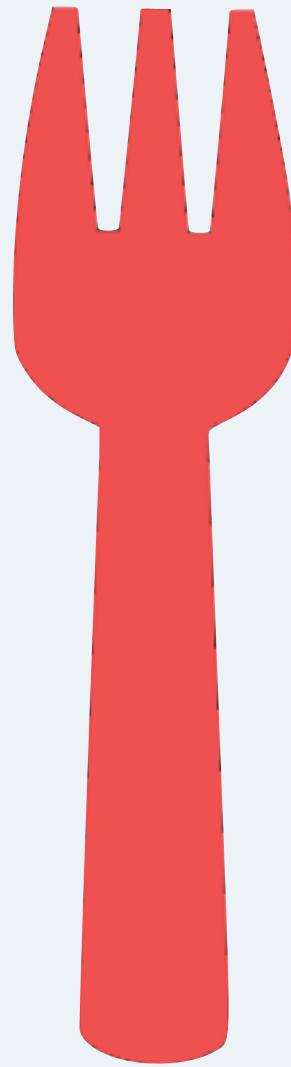


we know what can happen!



FORK WORKFLOWS





MAKES A COPY OF A REPOSITORY ON THE SERVER

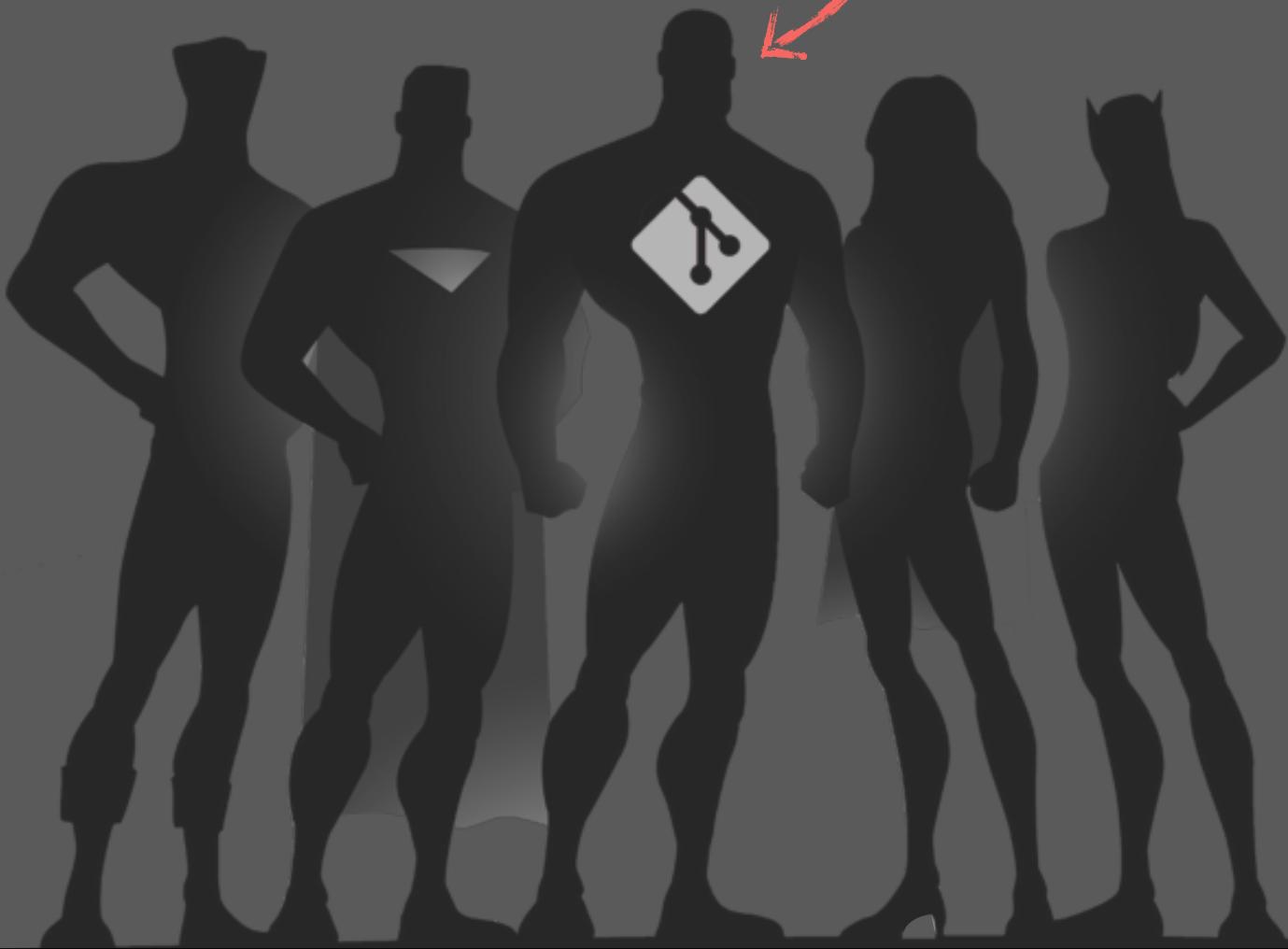
YOU CAN NOT WRITE DIRECTLY TO THE ORIGINAL REMOTE REPOSITORY

YOU CAN STILL GET CHANGES FROM THE ORIGINAL REMOTE REPOSITORY

forking rocks!

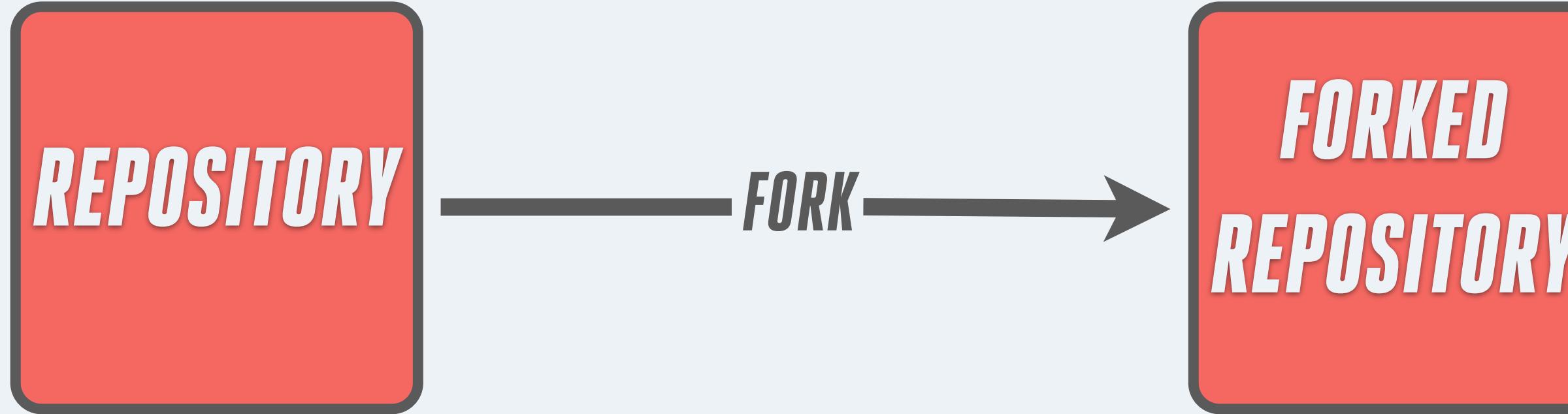
THE GATEKEEPER WORKFLOW

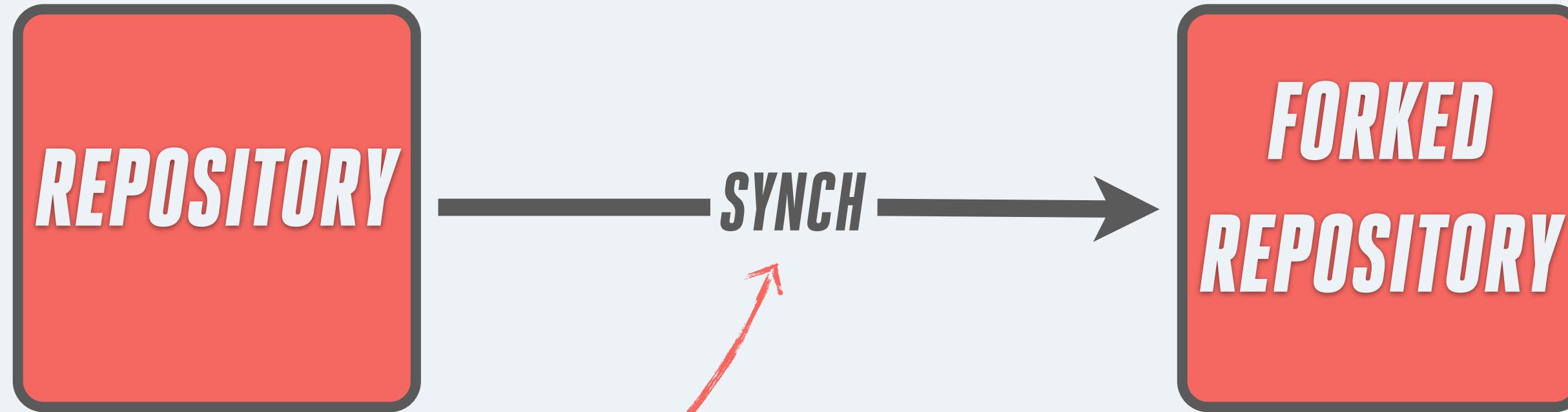
for teams





it's such an awesome Library

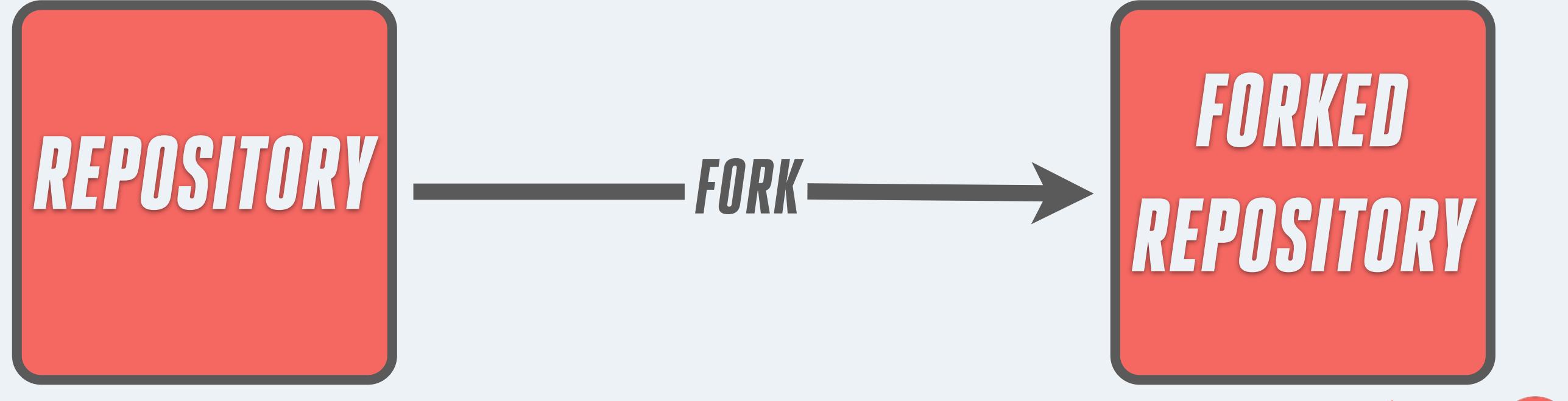




Get updates on request or automatically*

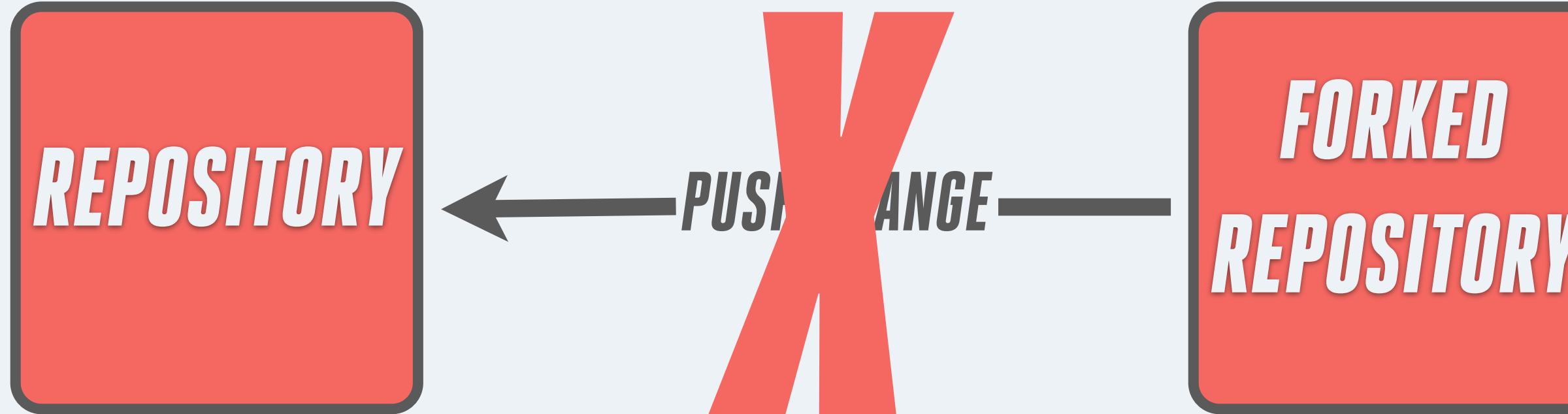


* with awesome tools like Stash or with Git hooks



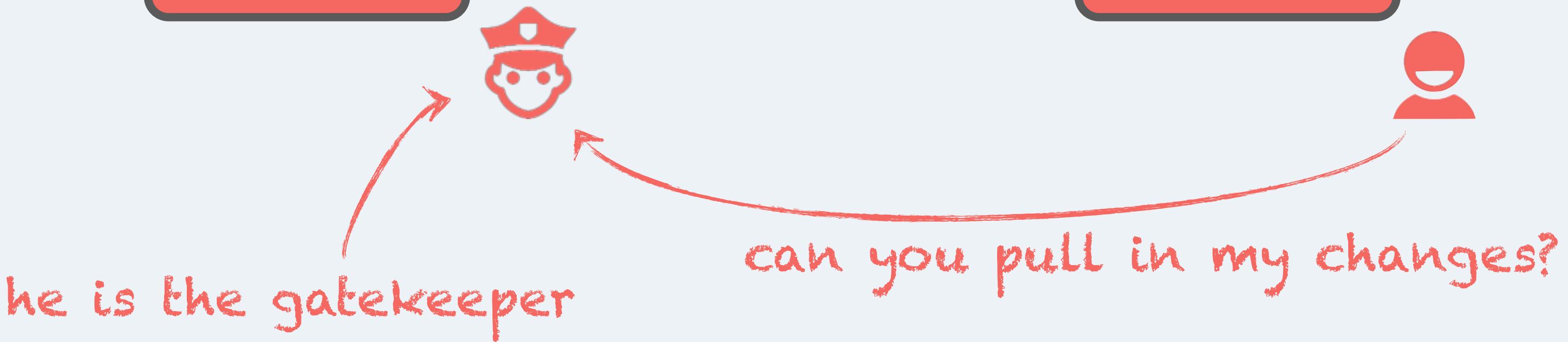
I've done an awesome change

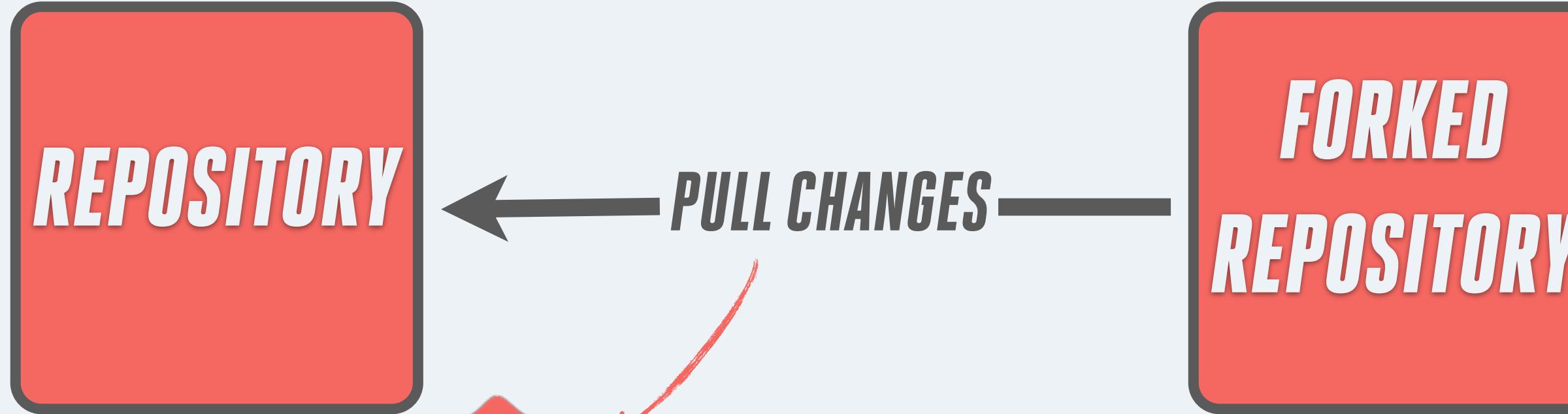




REPOSITORY

**FORKED
REPOSITORY**





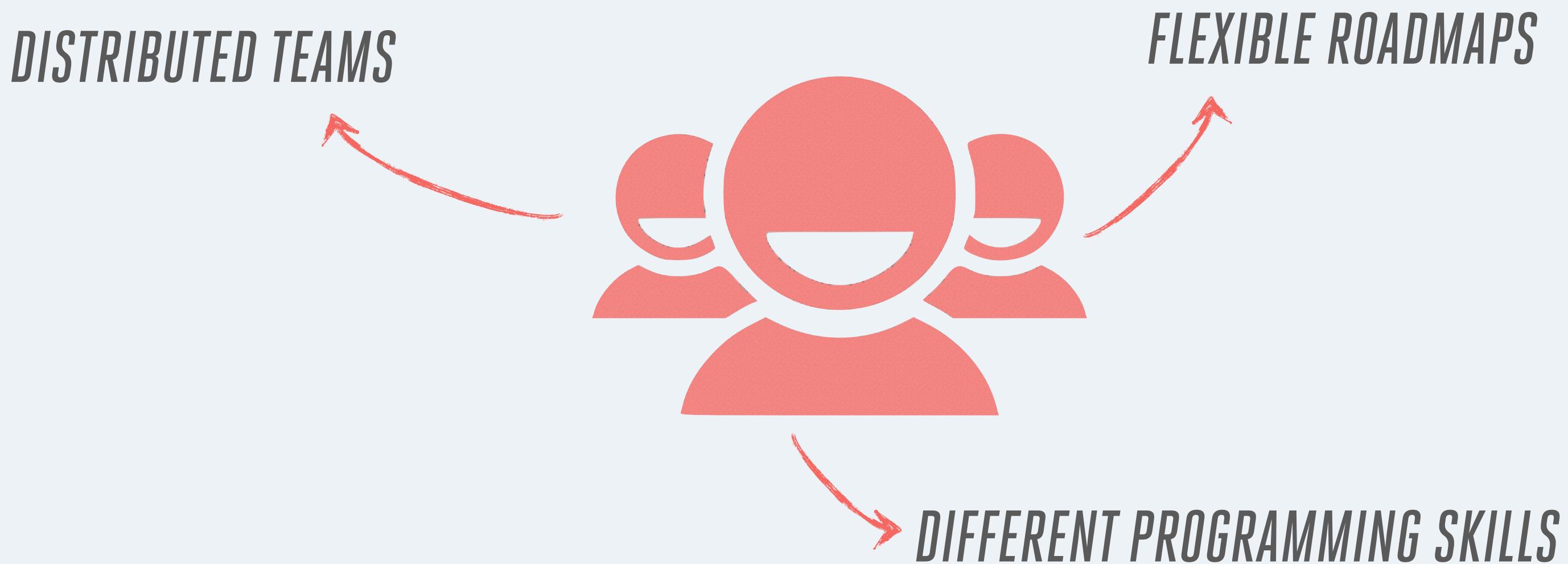
Likes the changes

LOOKS LIKE AN ***OPEN SOURCE*** PROCESS?



exactly and it works great!

LEARN FROM **OPEN SOURCE** TEAMS?





FORKING IN THE **ENTERPRISE**

4 REASONS FOR...



**FORKING IN THE
ENTERPRISE**



REASON 1

GREAT FOR CUSTOMIZING LIBRARIES

and still get bug fixes

A red arrow starts at the end of the sentence "and still get bug fixes" and curves upwards and to the left, pointing towards the top of the slide where the title "GREAT FOR CUSTOMIZING LIBRARIES" is located.



REASON 2

GREAT FOR INNOVATION SPIKES

A hand-drawn style red arrow starts from the bottom left and curves upwards and to the right, pointing towards the text "and maybe add it later".

and maybe add it later



REASON 3

but still be open for changes

PROTECTING YOUR COMPONENTS



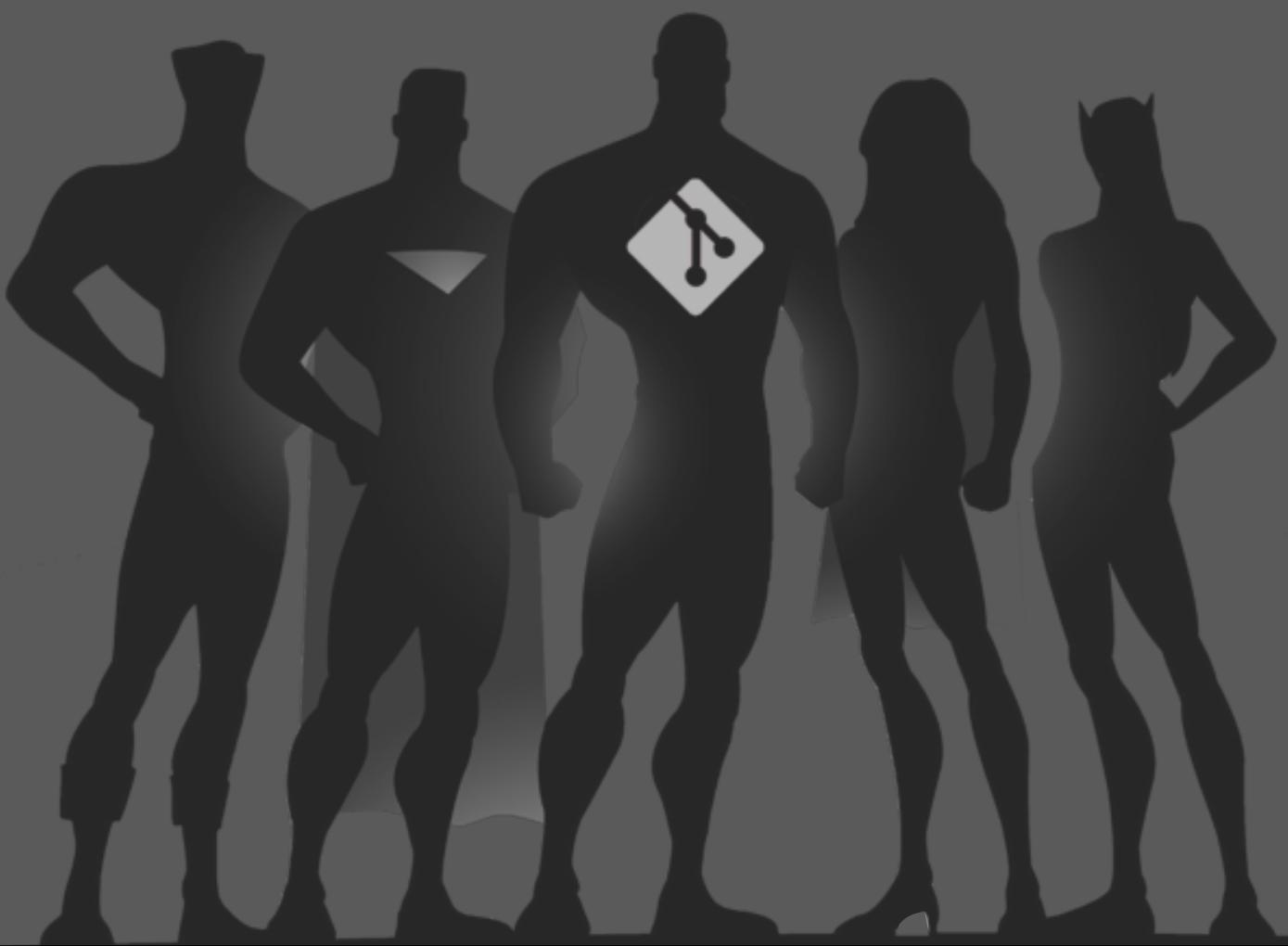
REASON 4

INTERACTION WITH CONTRACTORS & INTERNS

A hand-drawn style red arrow starts at the bottom left and curves upwards and to the right, ending near the word "protect".

protect your sources

THE DICTATORS & LIEUTENANTS WORKFLOW





an army of developer

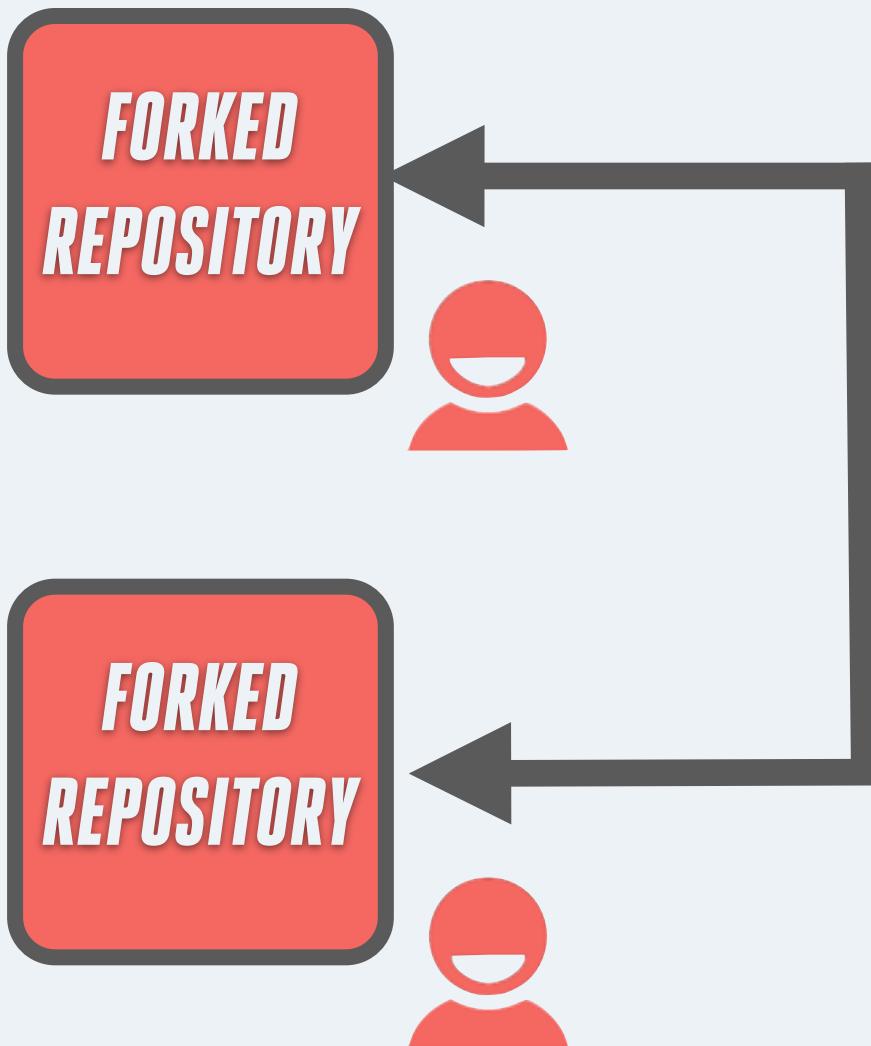


a few
integrator



and a friendly
dictator

**BLESSED
REPOSITORY**



**FORKED
REPOSITORY**



**FORKED
REPOSITORY**

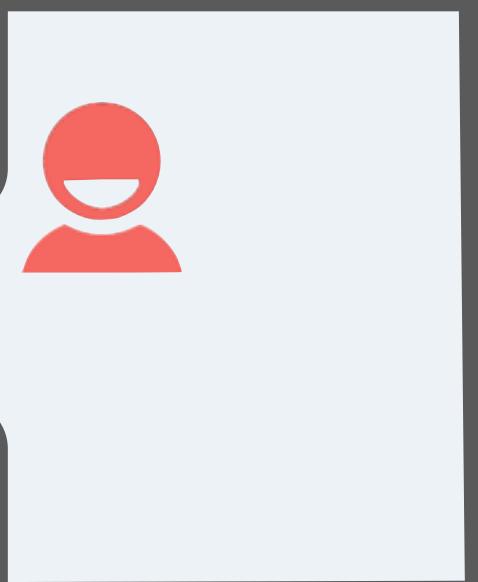


Pull

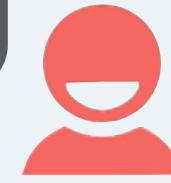
**BLESSED
REPOSITORY**



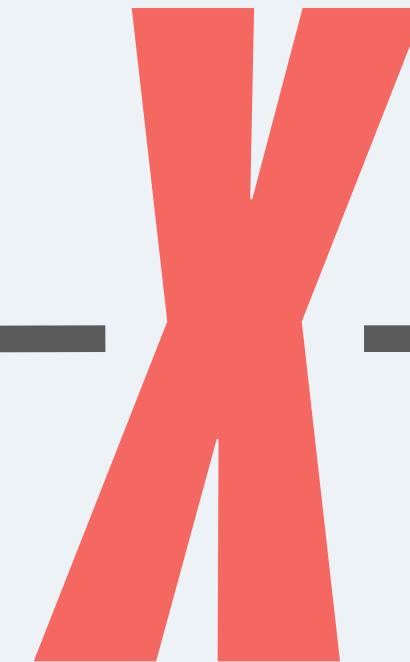
**FORKED
REPOSITORY**



**FORKED
REPOSITORY**

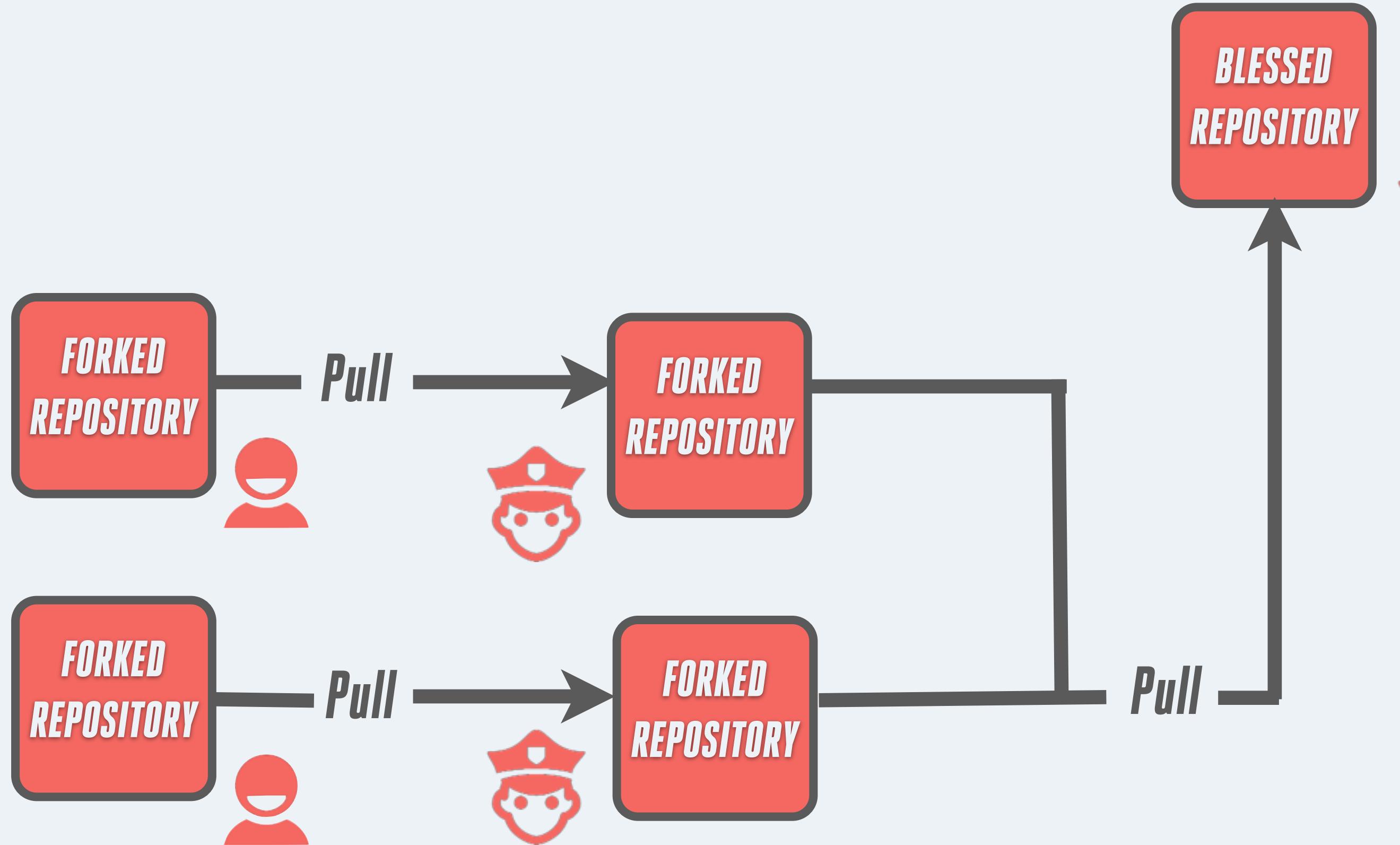


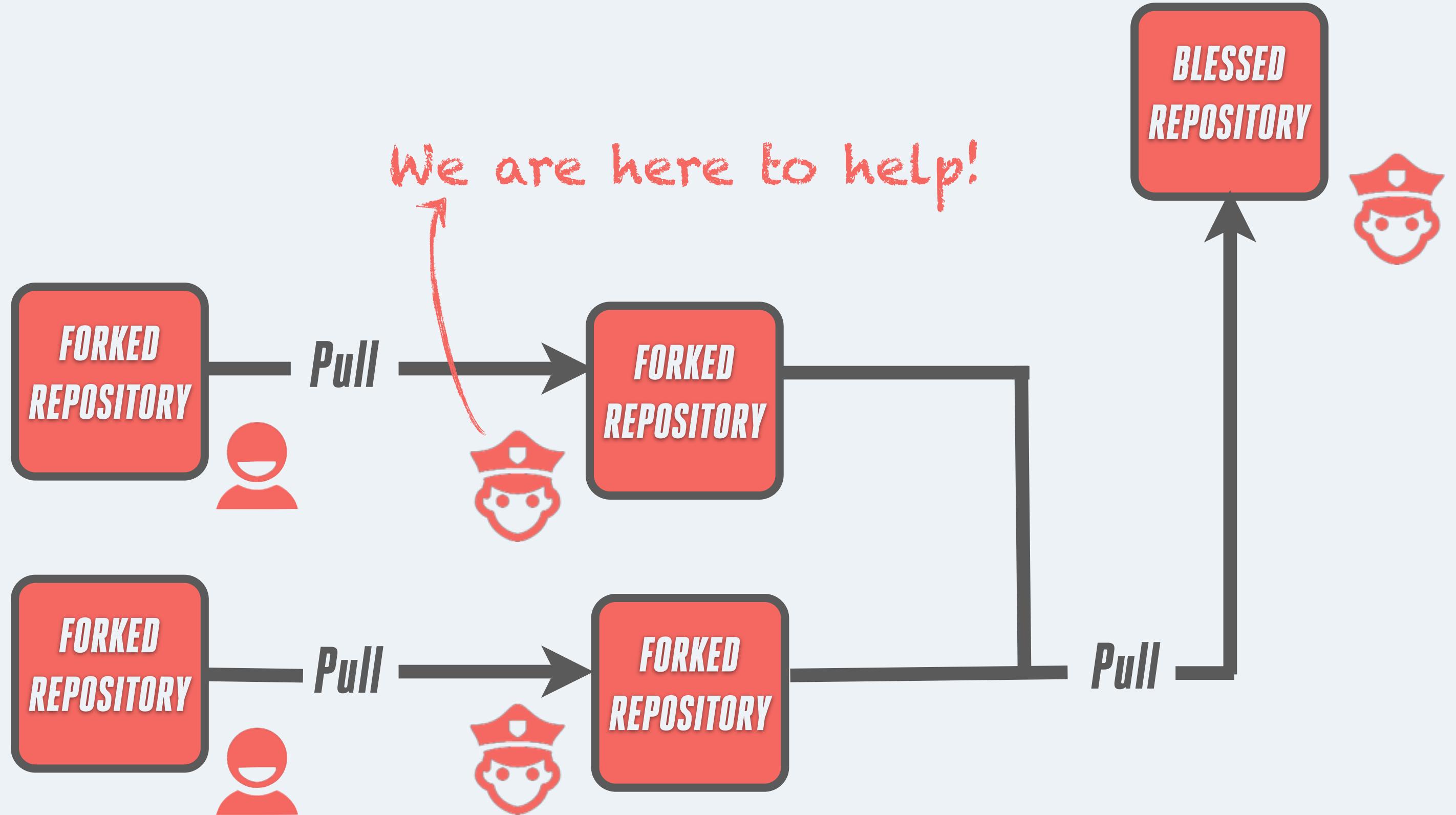
too much stuff! ↩



**BLESSED
REPOSITORY**









GREAT FOR HUGE PROJECTS

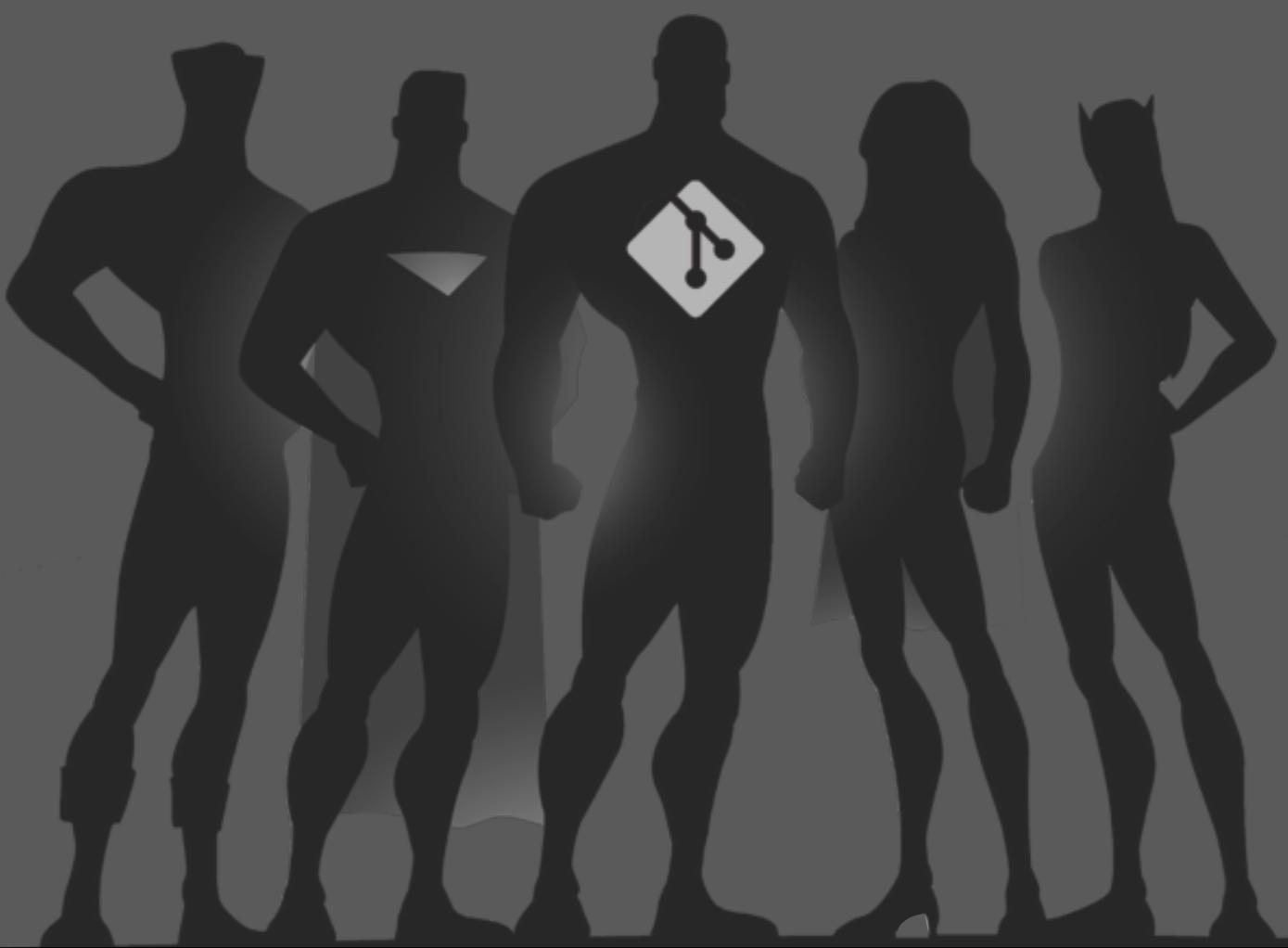
THIS IS A POTENTIAL BOTTLENECK



**YOU NEED MORE FREEDOM?
AND WANT TO MOVE FAST?**



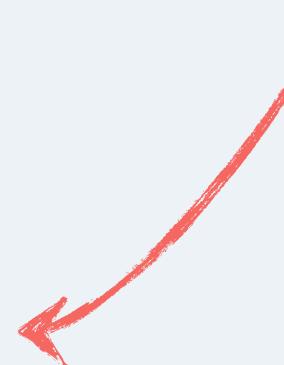
THE CENTRALIZED WORKFLOW



HELLO SUBVERSION USERS!

**THIS
IS
YOUR *FLOW***

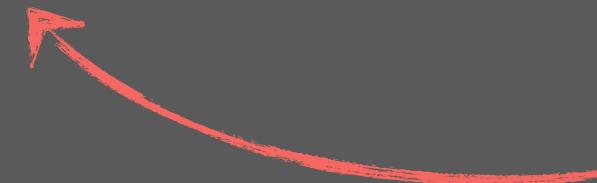
**but Git makes
it even better**



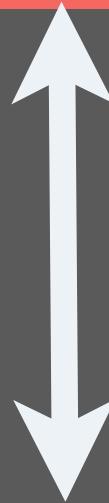
MASTER

**ONE BRANCH
TO RULE THEM ALL**

all code goes in here

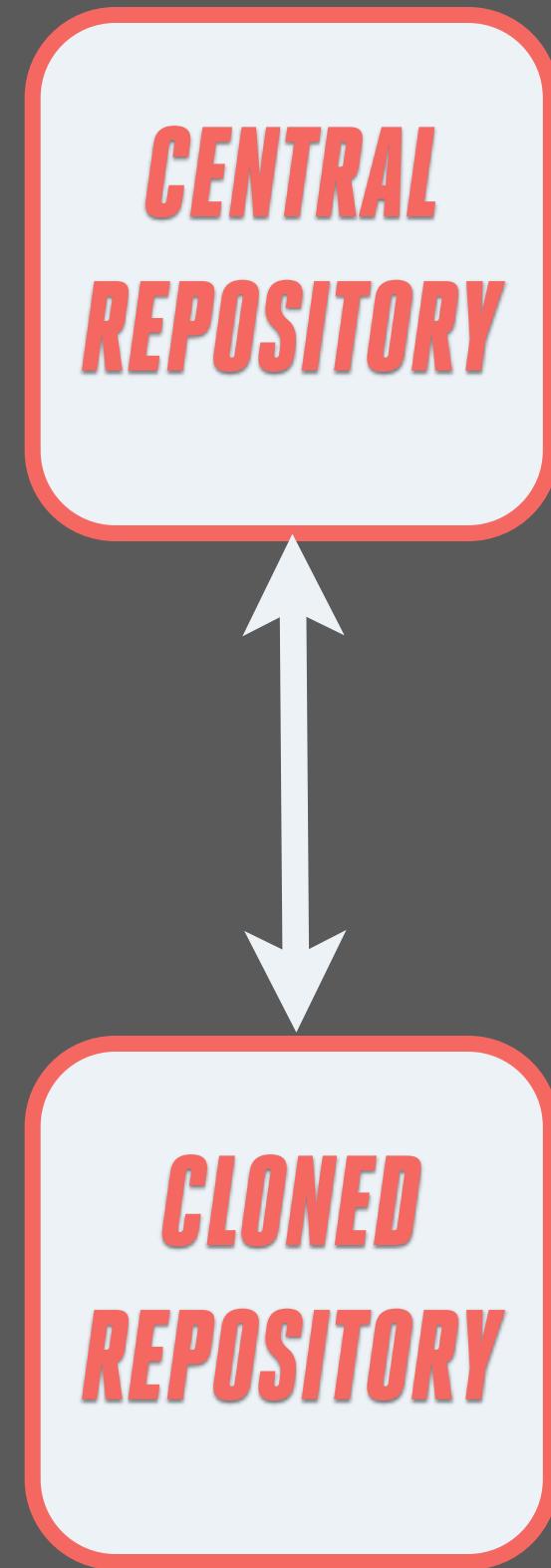


**CENTRAL
REPOSITORY**



**CLONED
REPOSITORY**

your (full) local copy for
development



**CENTRAL
REPOSITORY**

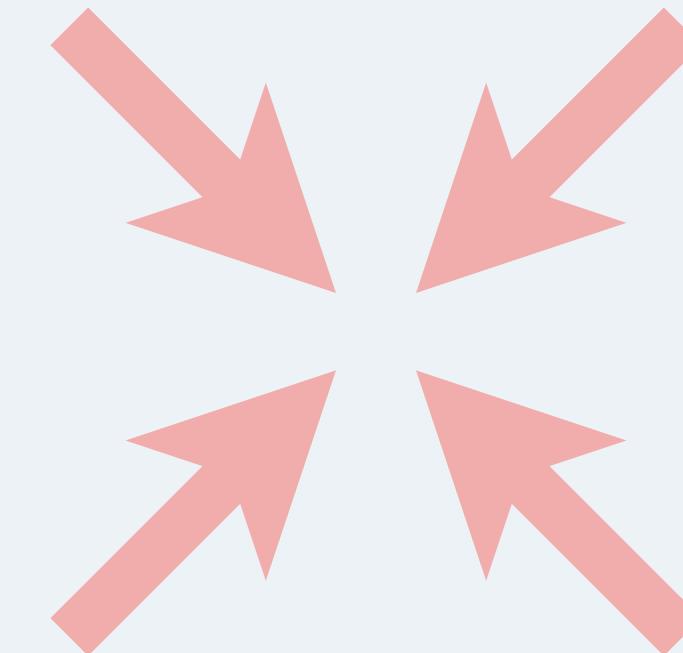


**CLONED
REPOSITORY**

do that often!

SIMPLE & EASY TO LEARN

CONFLICTS GET VISIBLE EARLY

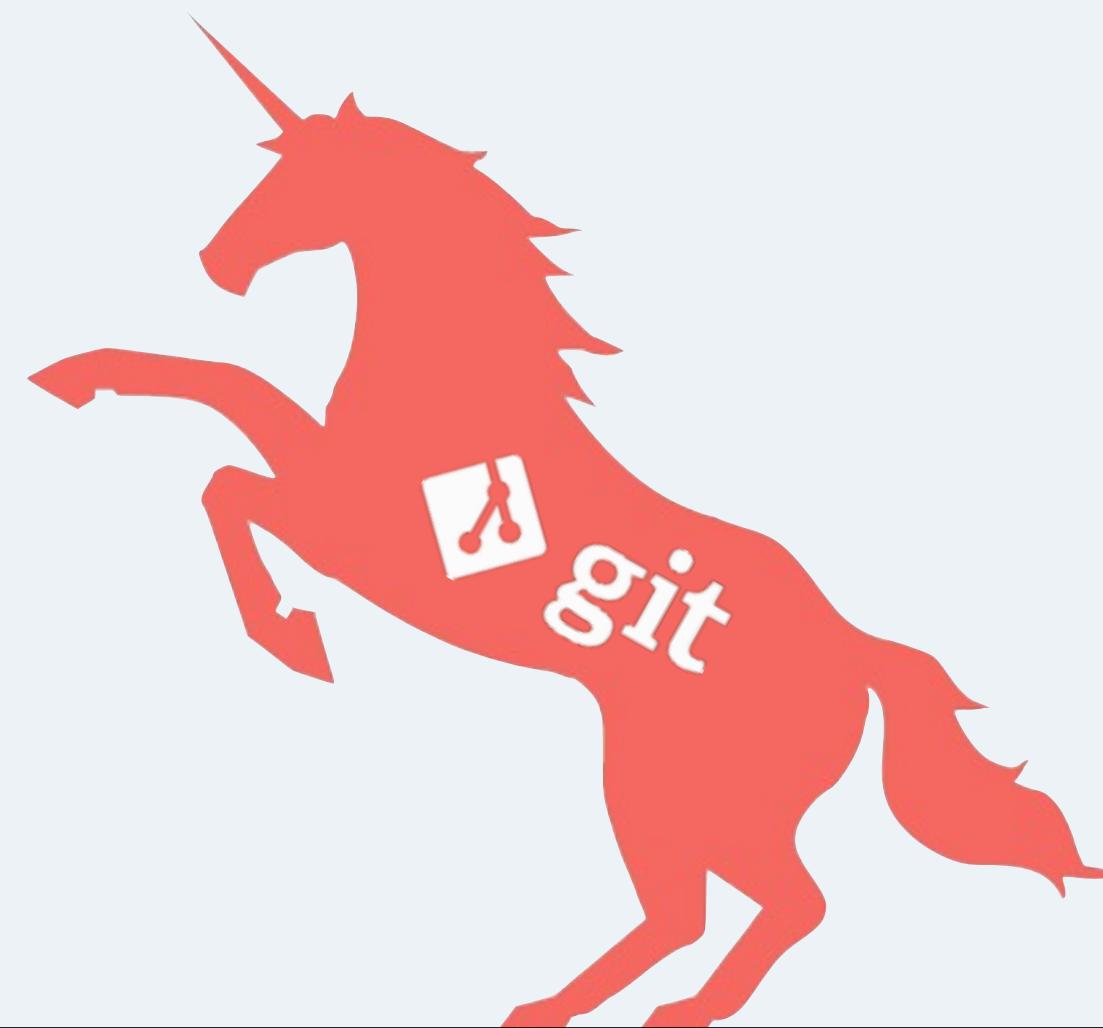
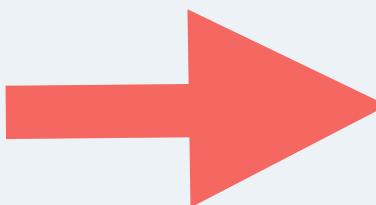


+CI = ALWAYS DEPLOYABLE

USED FOR YEARS

GREAT FOR MIGRATING

don't change too much at once



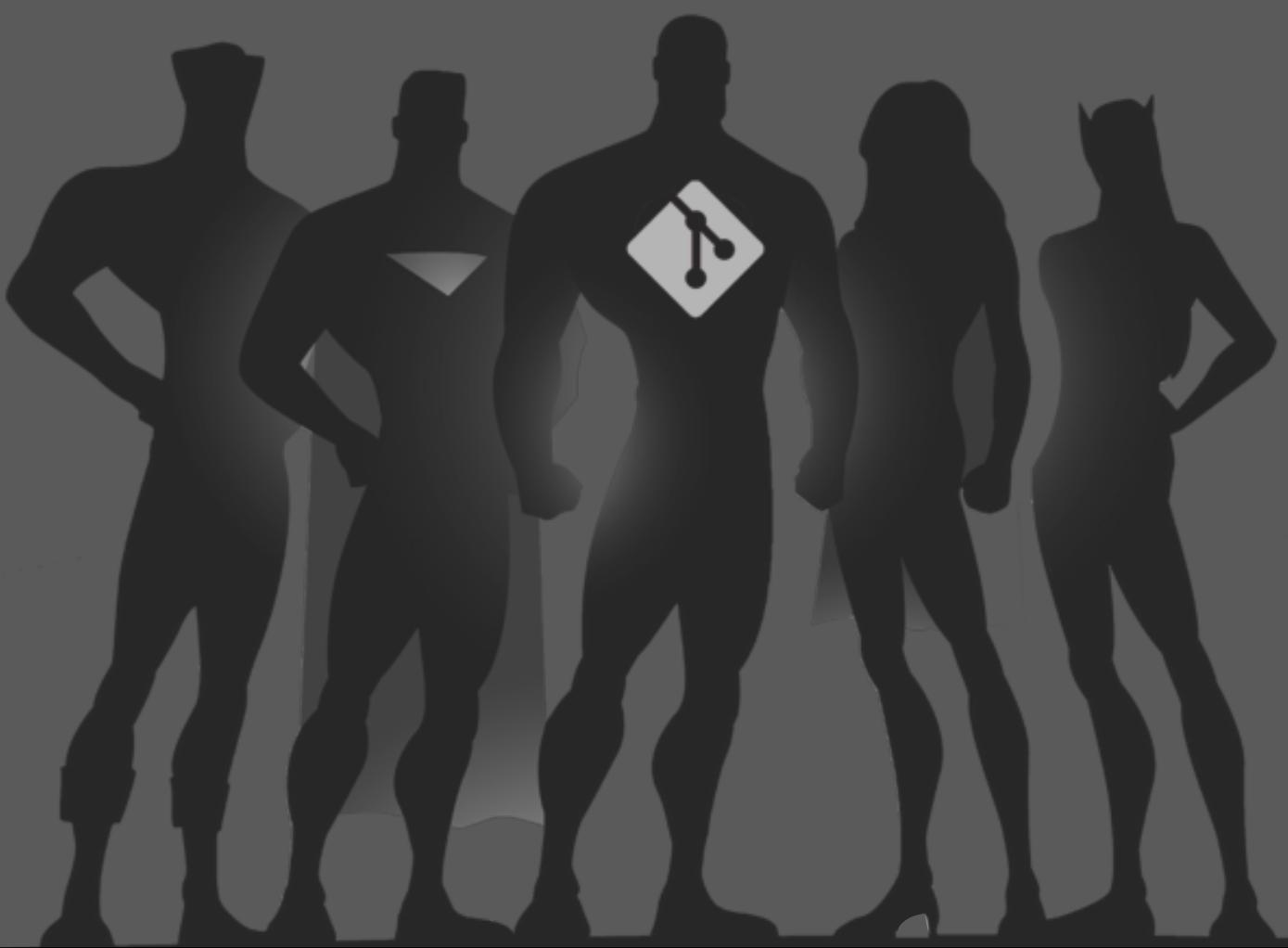
UNFINISHED FEATURES



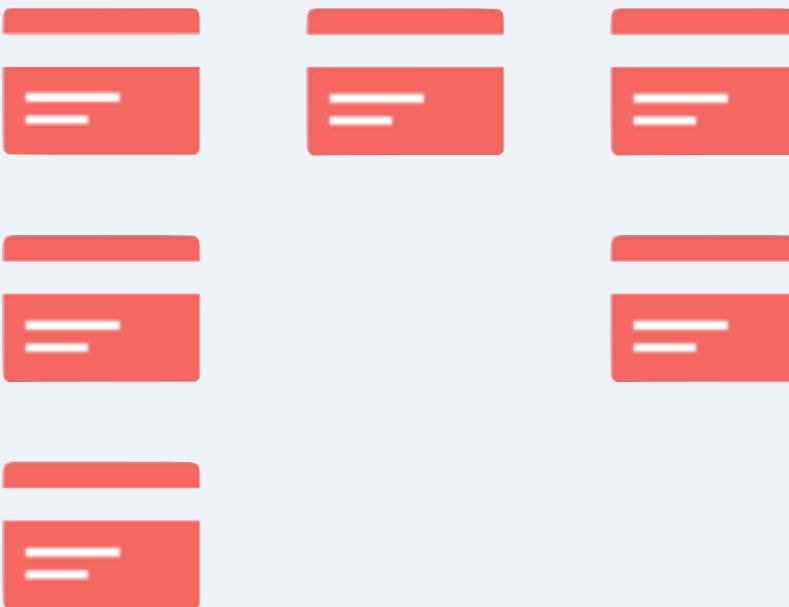
FEATURE TOGGLES



THE FEATURE BRANCH WORKFLOW



WE ORGANIZE WORK IN TASKS, STORIES, FEATURES



WHY NOT HAVE THAT IN CODE, TOO?

MASTER

FEATURE 2

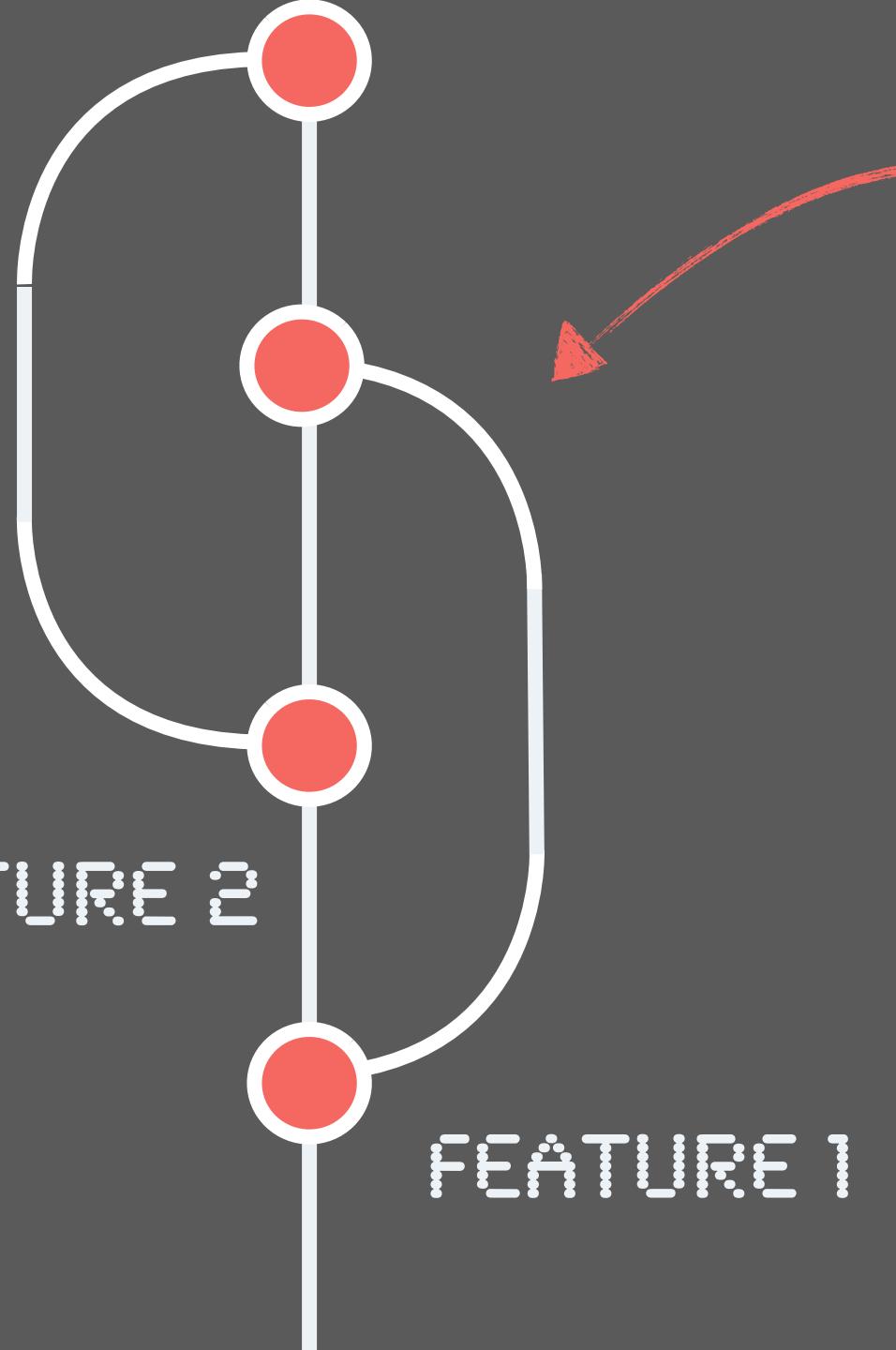
FEATURE 1



BRANCH PER FEATURE

1 to X devs working on a branch

MASTER



BRANCH PER FEATURE

get closed when done

CLEAN MASTER BRANCH



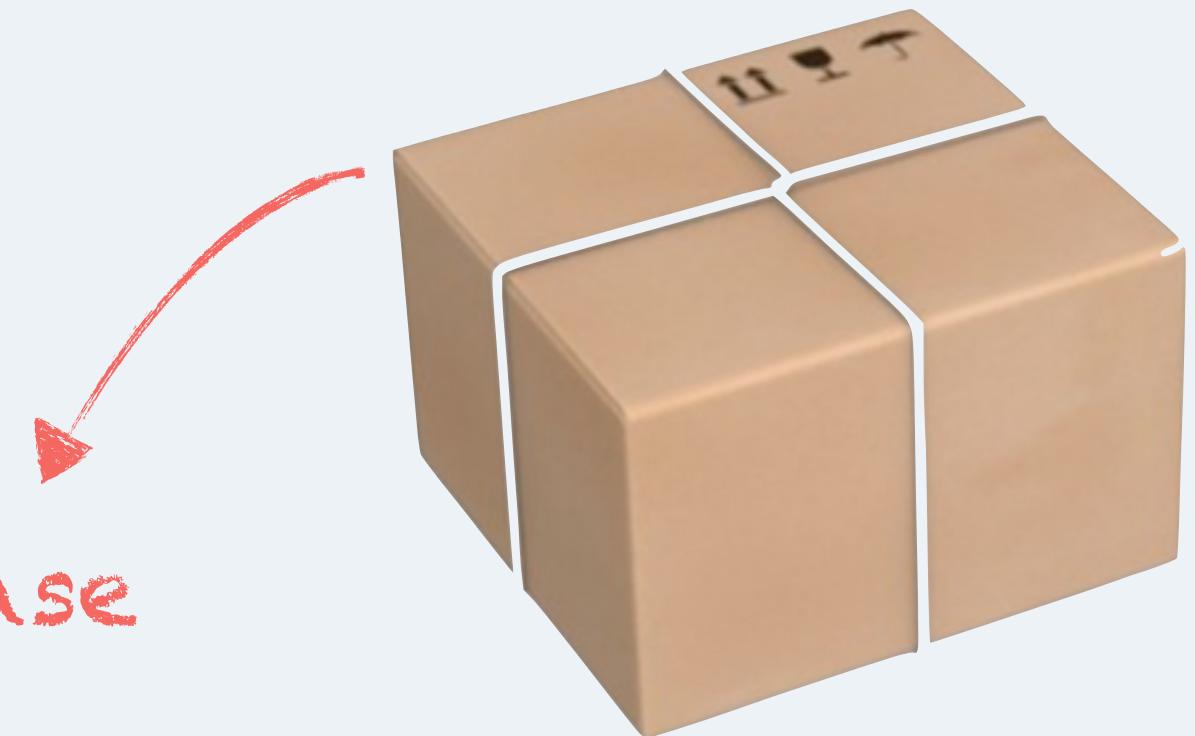
ISOLATED FEATURE DEVELOPMENT



GREAT

FOR FIXED TIME / FIXED FEATURE DEVELOPMENT

develop while finishing for release



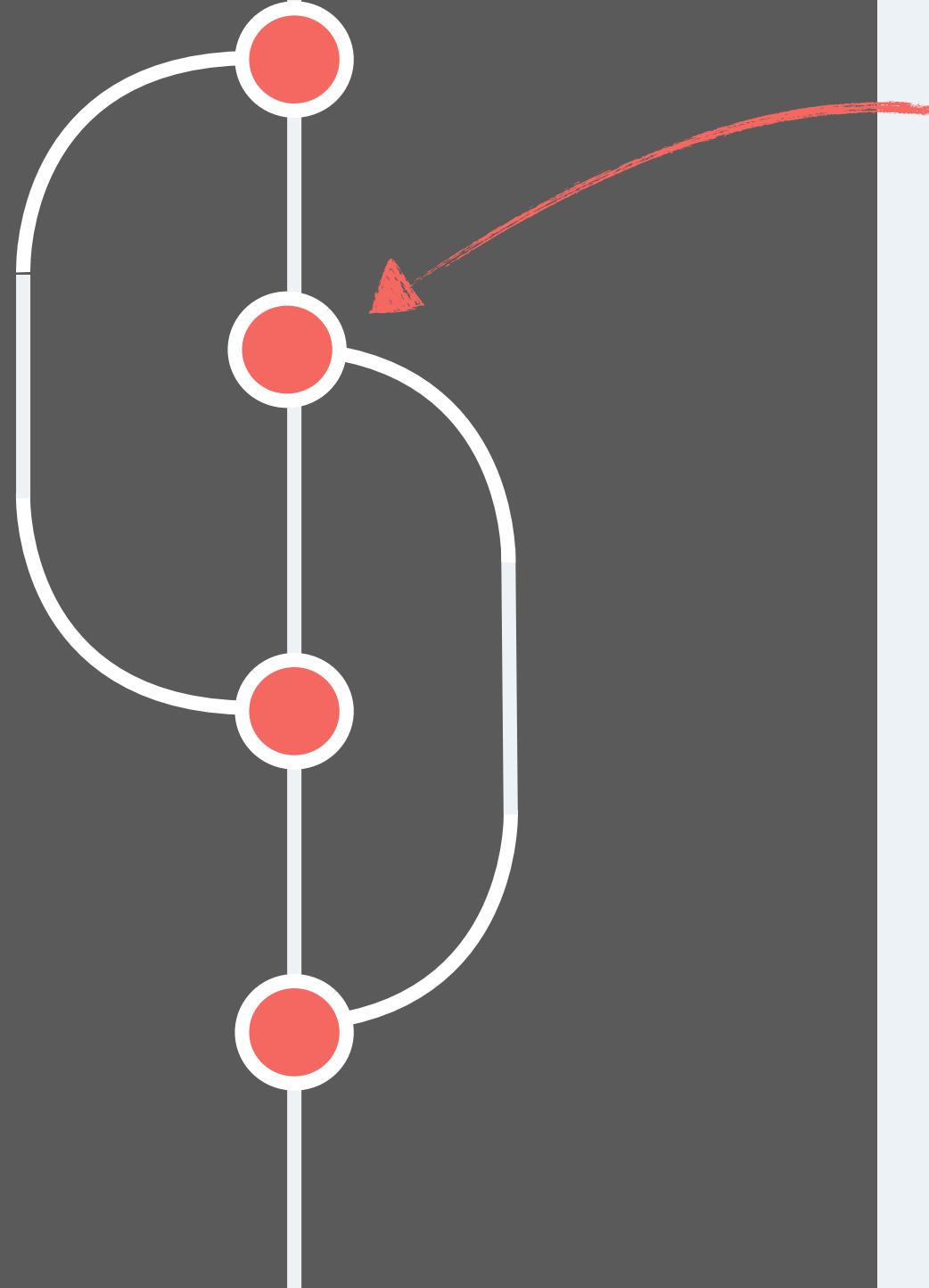
GREAT

FOR CONTINUOUS DELIVERY



deliver when done

MASTER

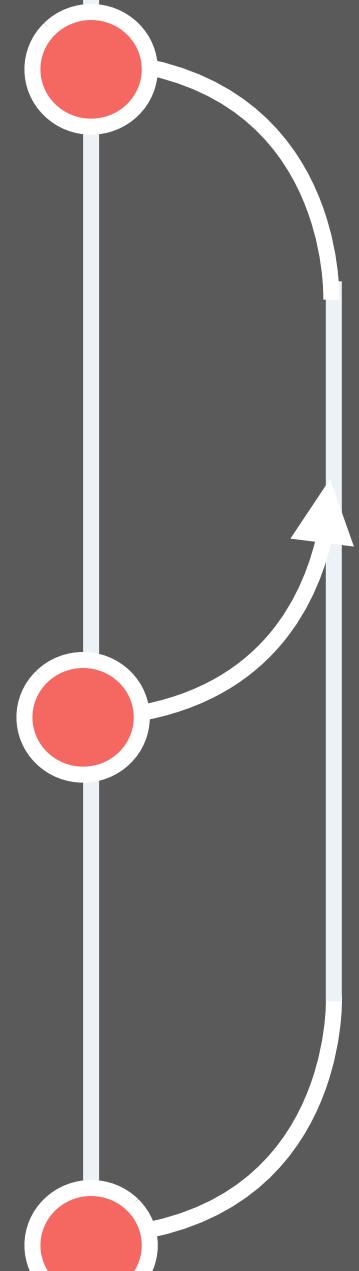


FEATURE BRANCH TIPS

merge frequently



MASTER

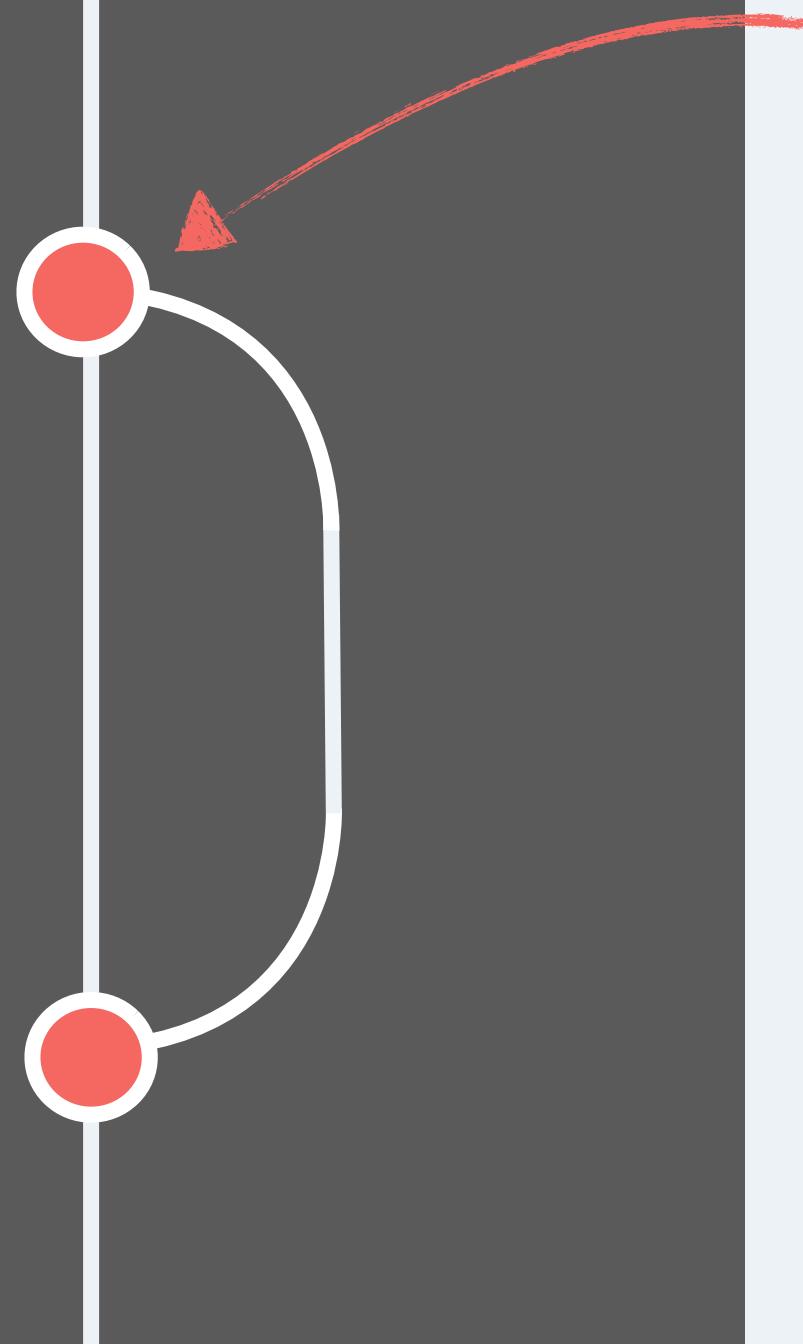


FEATURE BRANCH TIPS

synch with MASTER frequently



MASTER



FEATURE BRANCH TIPS

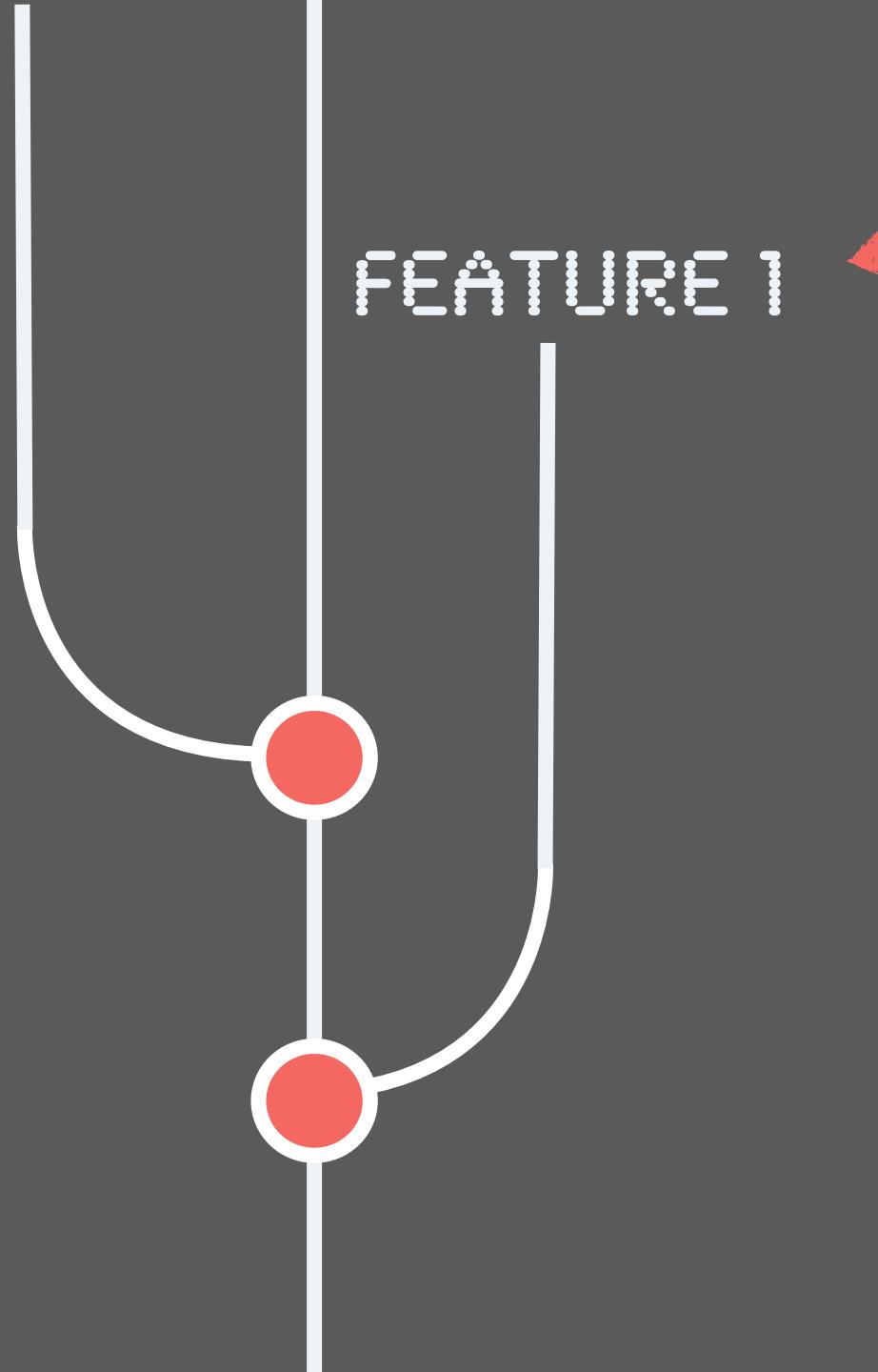
delete branch when done



MASTER

FEATURE 2

FEATURE 1



BRANCH PER FEATURE

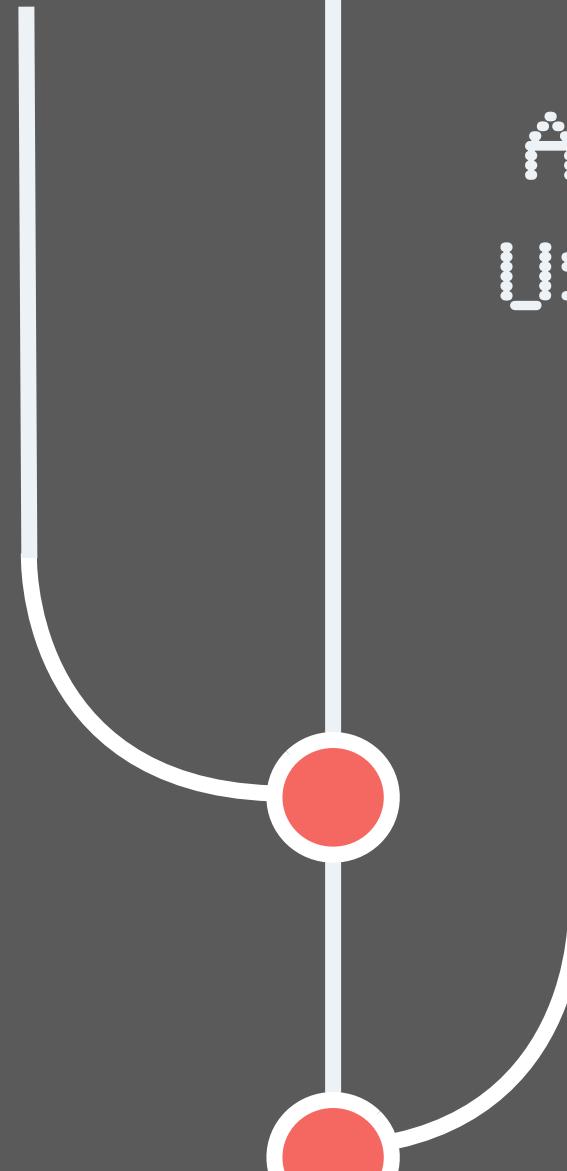
useful branch names



MASTER

DELETE
USER

ADD
USER



BRANCH PER FEATURE

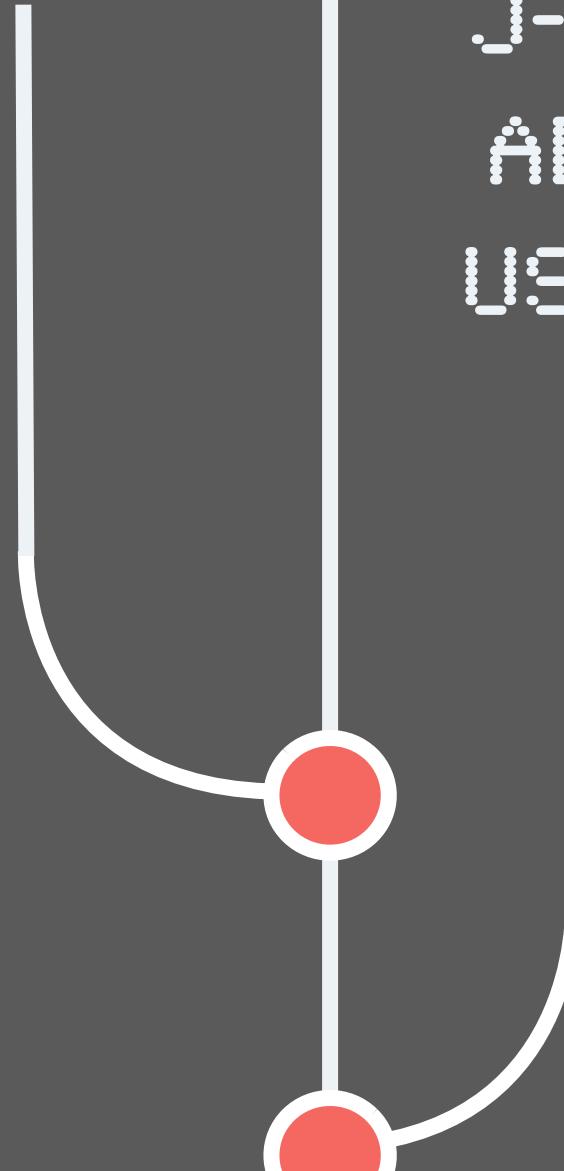
useful branch names



J-98
DELETE
USER

MASTER

J-68
ADD
USER



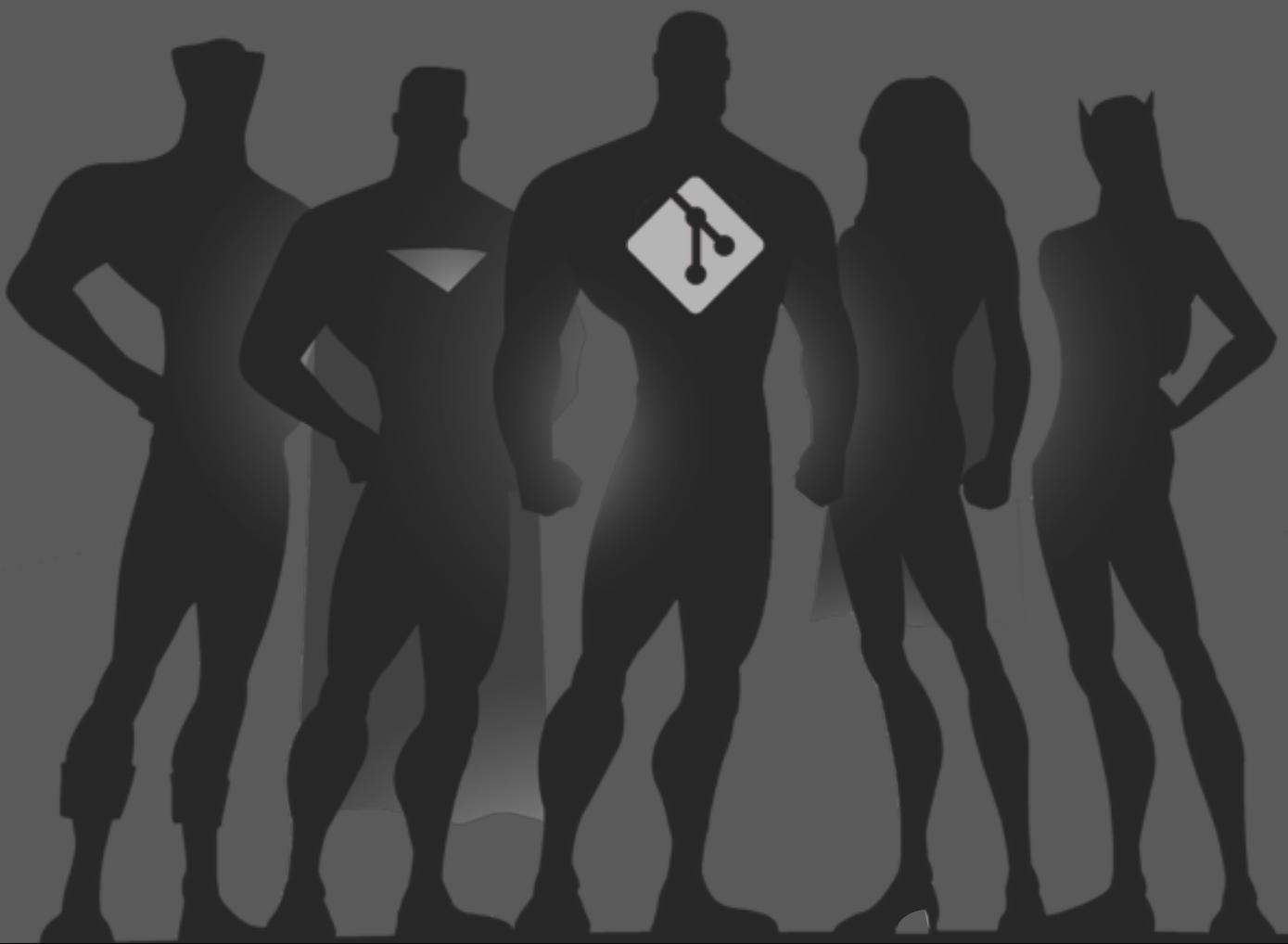
use key from issue tracker



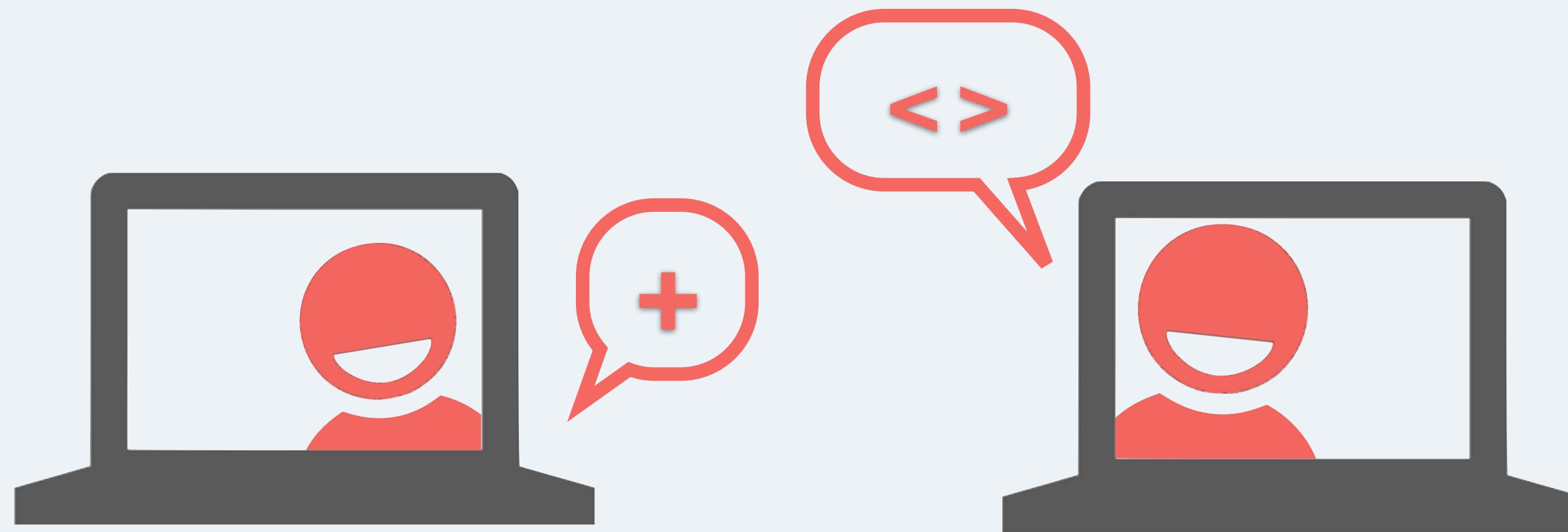
BRANCH PER FEATURE

THE FEATURE BRANCH WORKFLOW

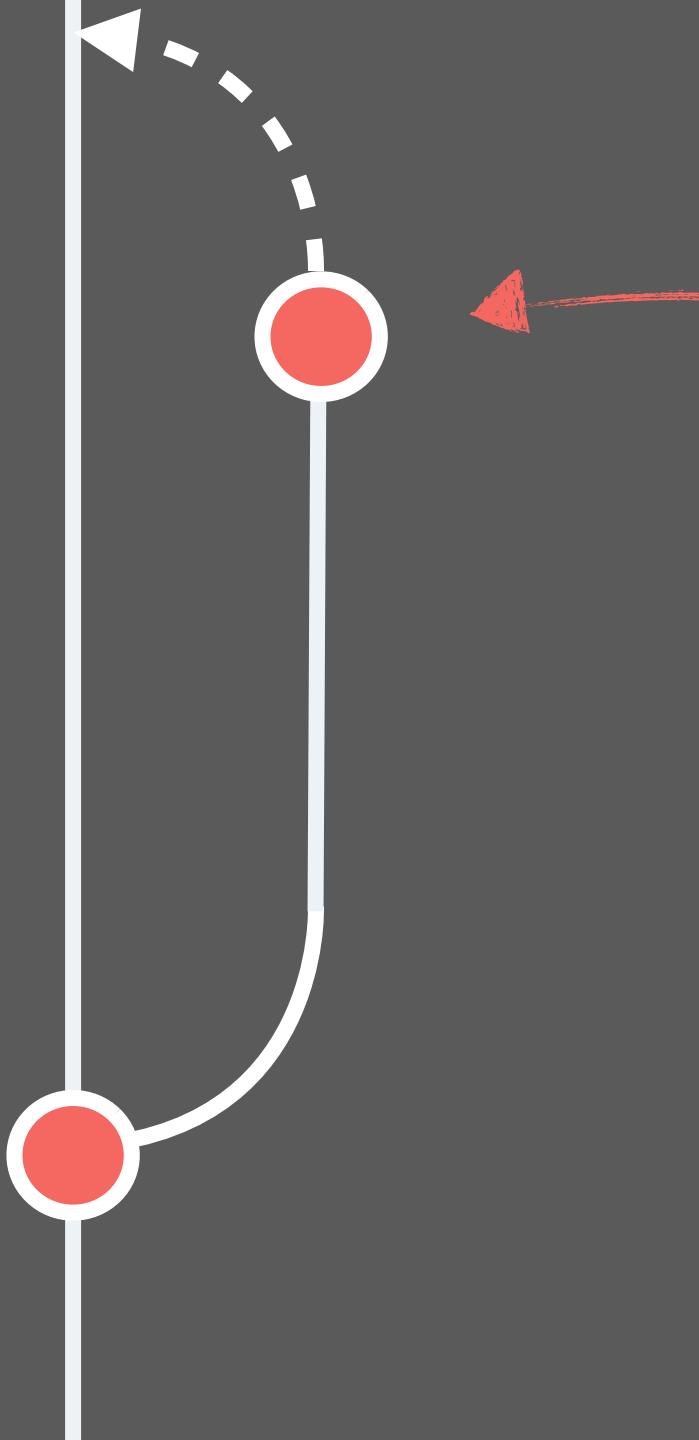
with code reviews



PULL REQUESTS

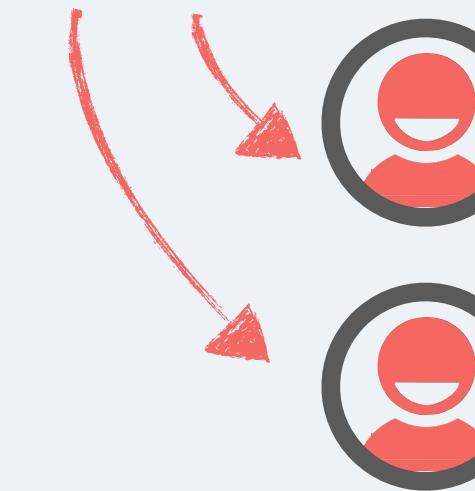


MASTER

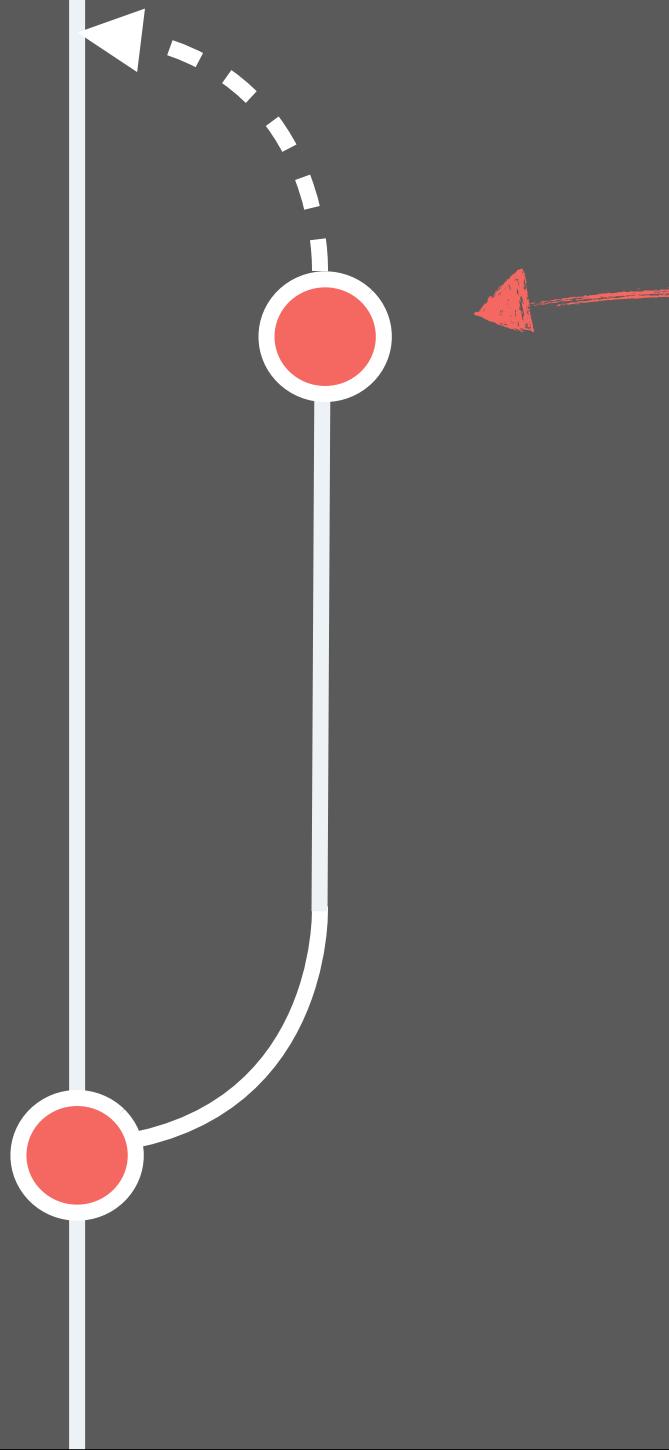


Pull Request

what do you think?



MASTER

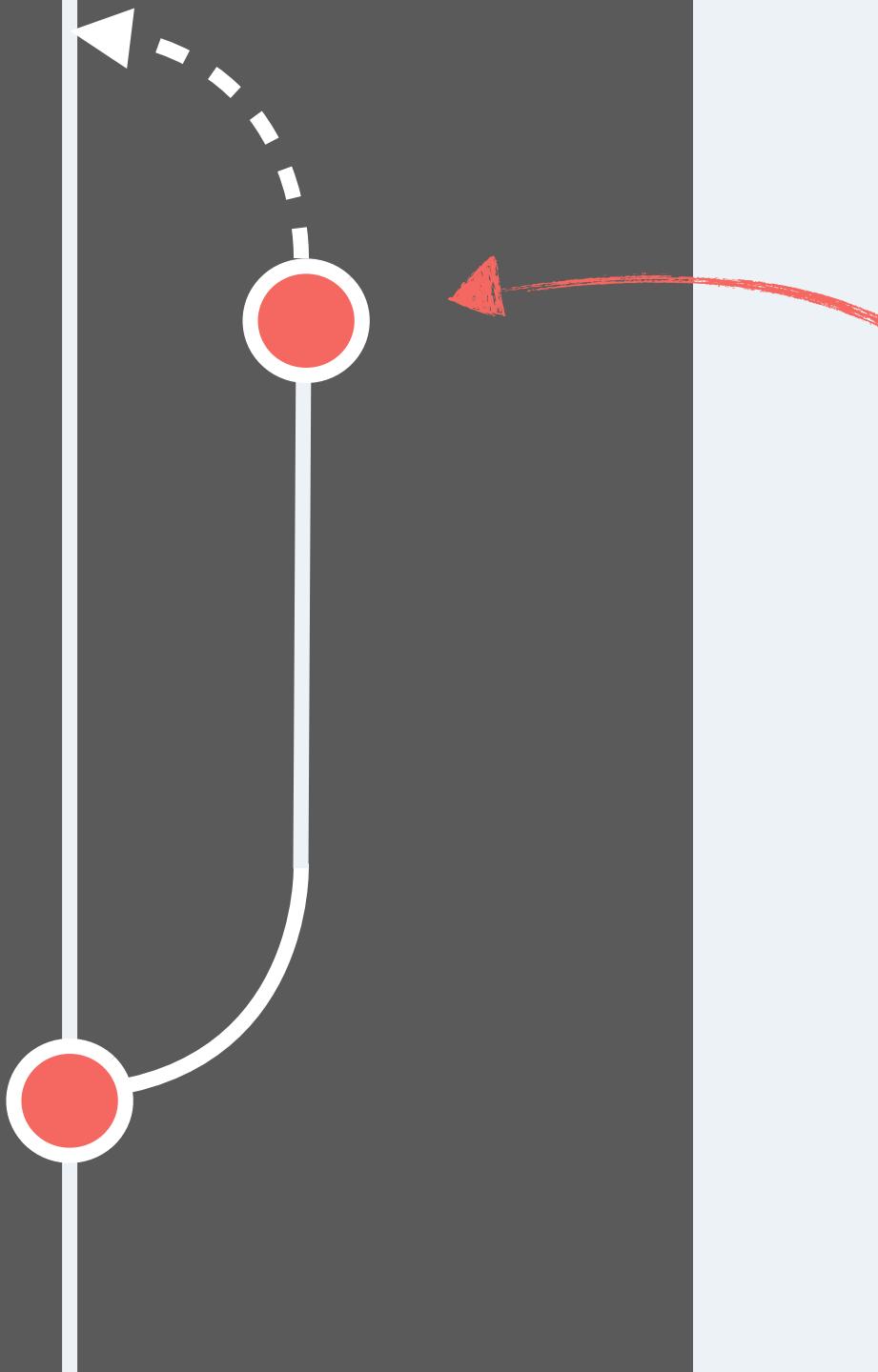


Pull Request

discuss changes

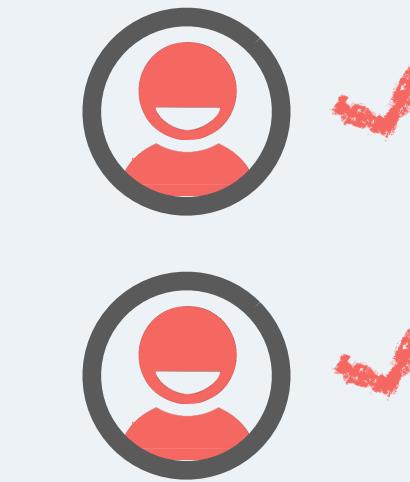


MASTER

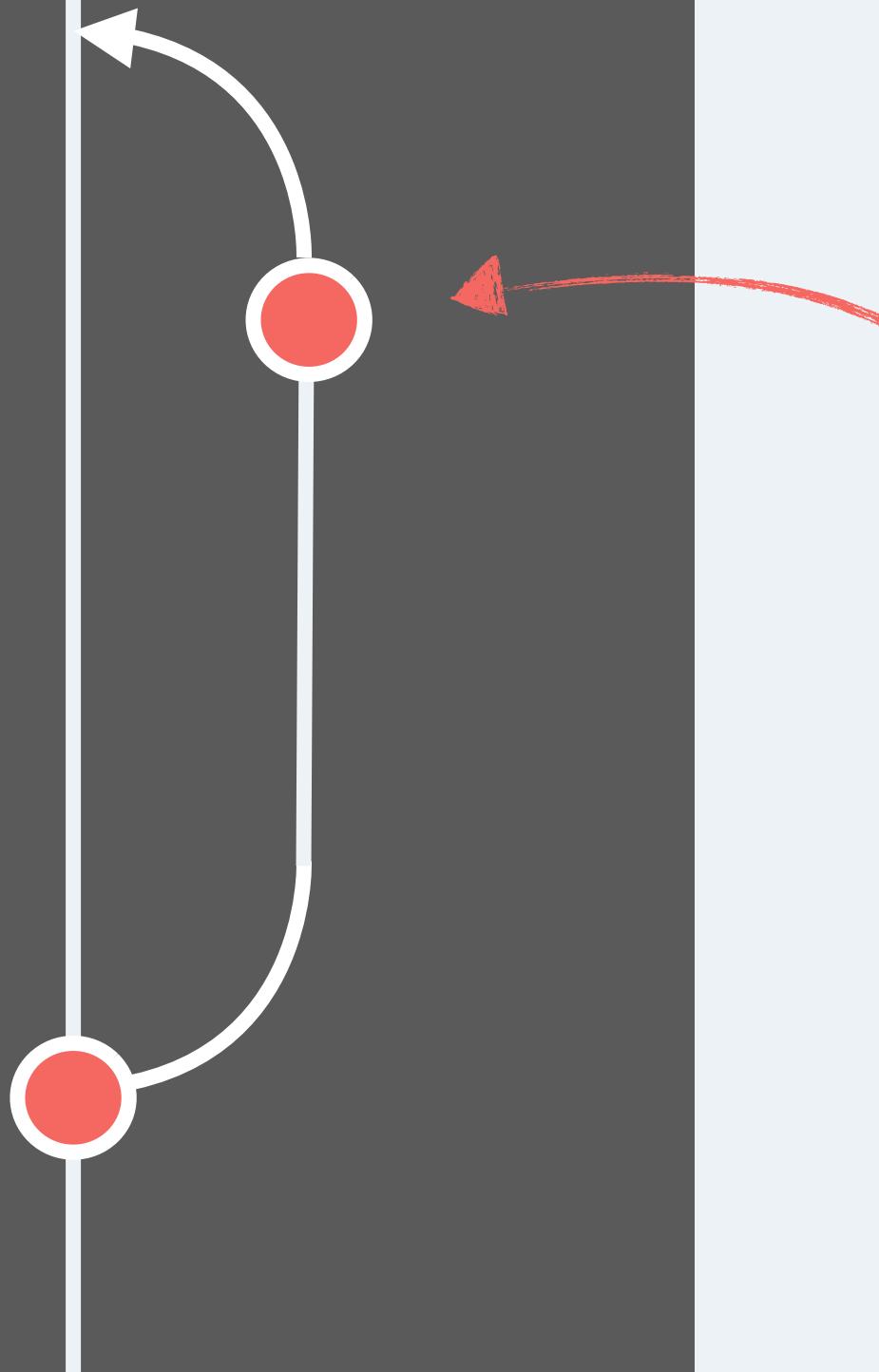


Pull Request

approve changes



MASTER



Pull Request

THE



WORKFLOW *

***SOURCE: ZACH HOLMAN "HOW GITHUB WORKS"**



MORE STABILITY

MASTER

DEVELOP

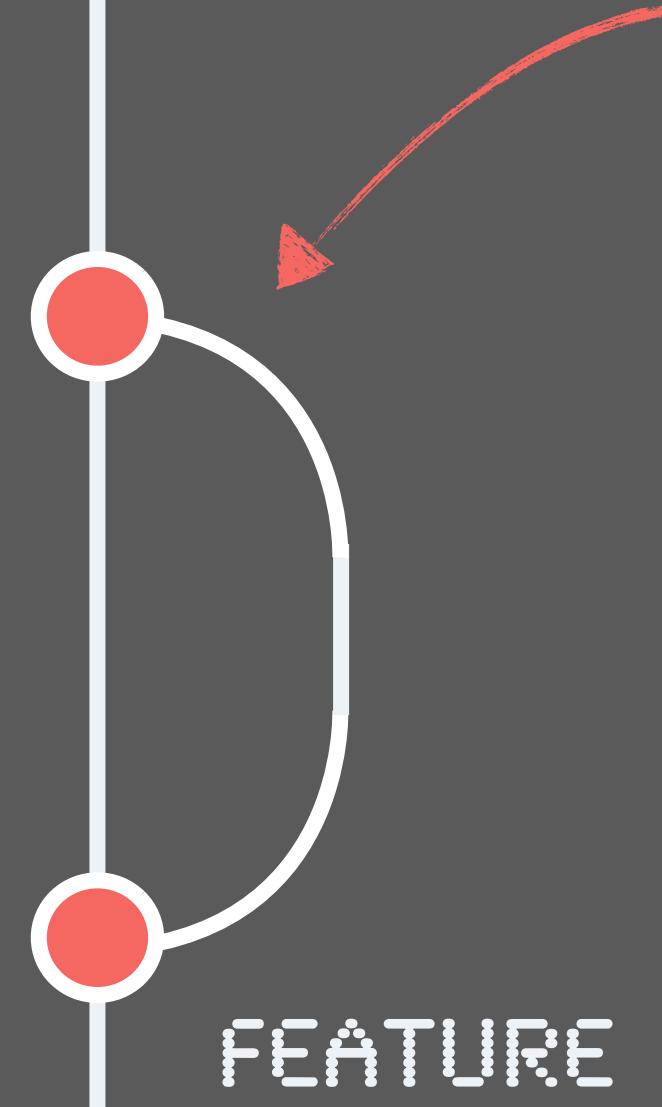


STABILITY

new branch called DEVELOP

MASTER

DEVELOP

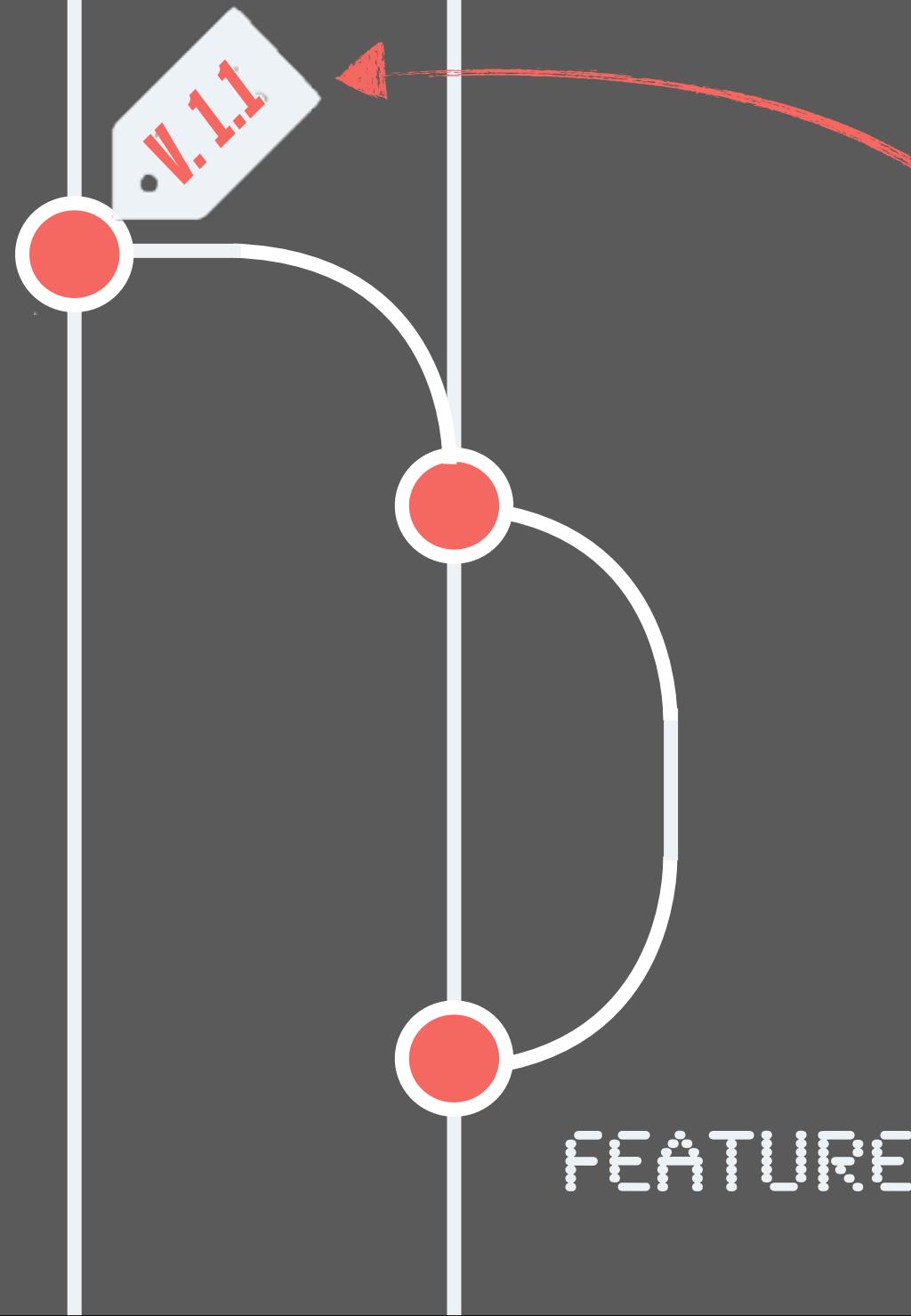


STABILITY

we integrate changes here

MASTER

DEVELOP

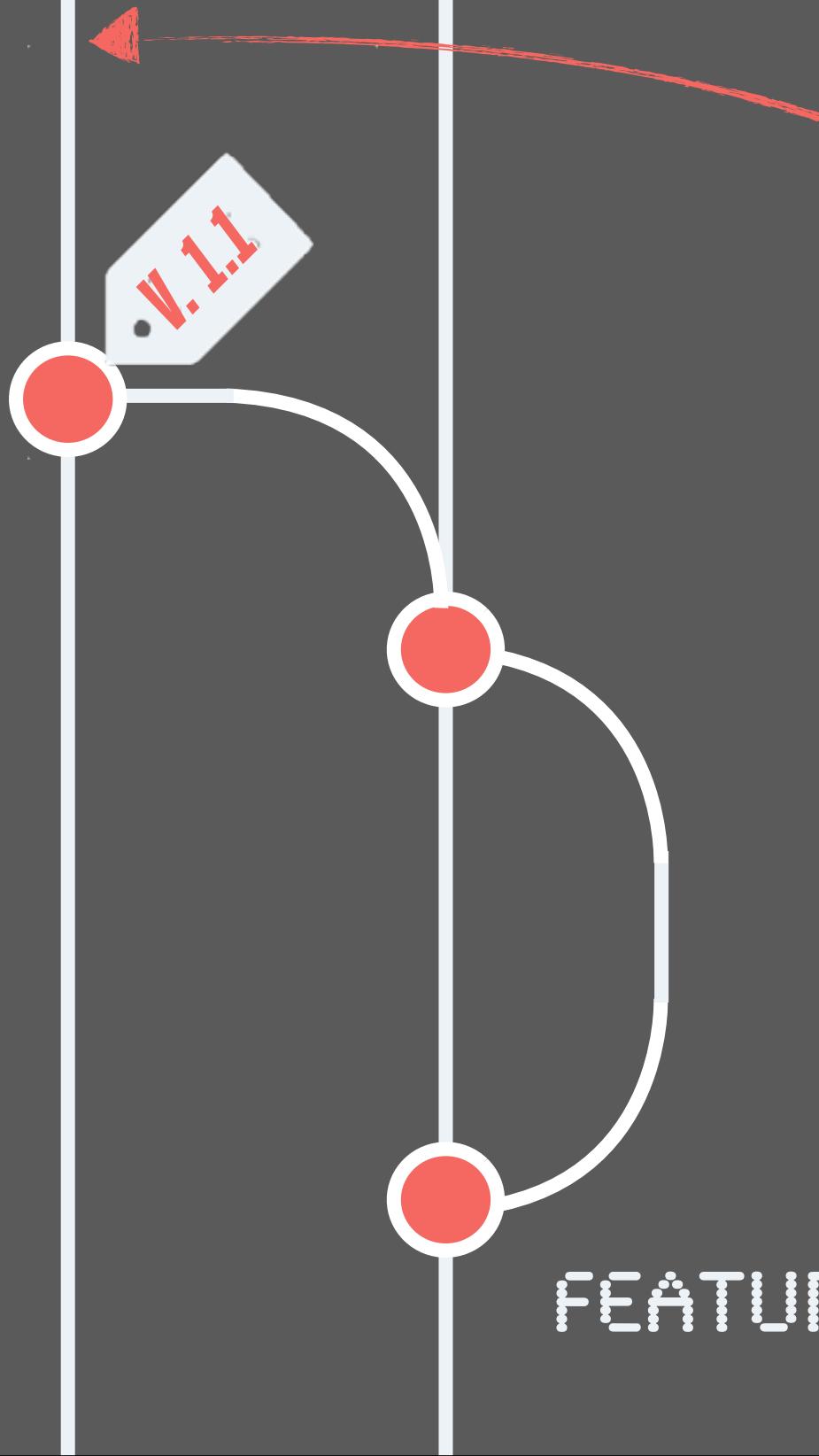


STABILITY

and tag our releases here

MASTER

DEVELOP



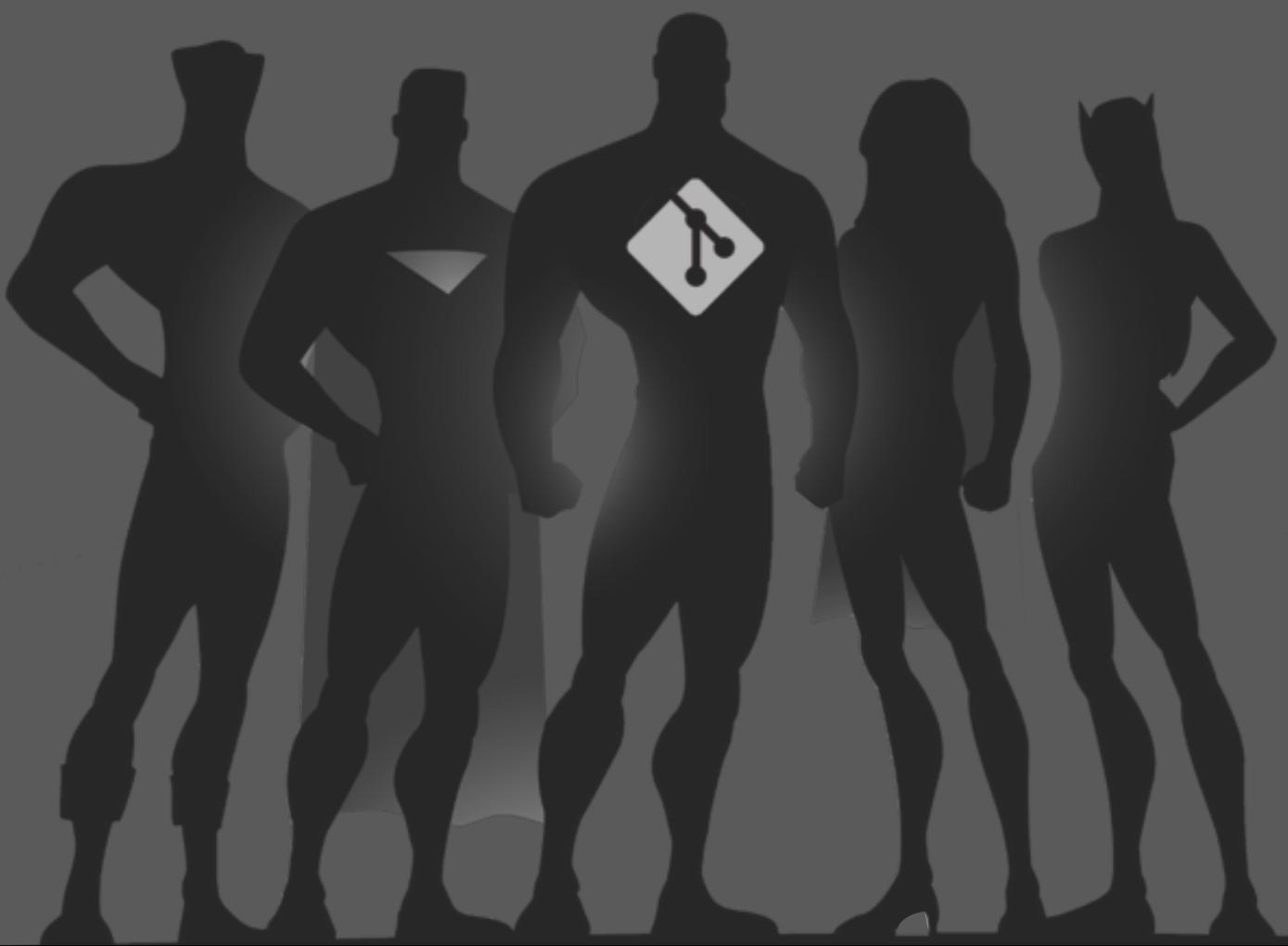
STABILITY

makes this branch stable

THE

Bitbucket WORKFLOW

THE RELEASE WORKFLOW





COOL STUFF
WE'RE FLOWING!



inspect your release candidate

RELEASE CANDIDATES



inspect your release candidate
...and don't destroy your flow

RELEASE CANDIDATES

RELEASE



RELEASE

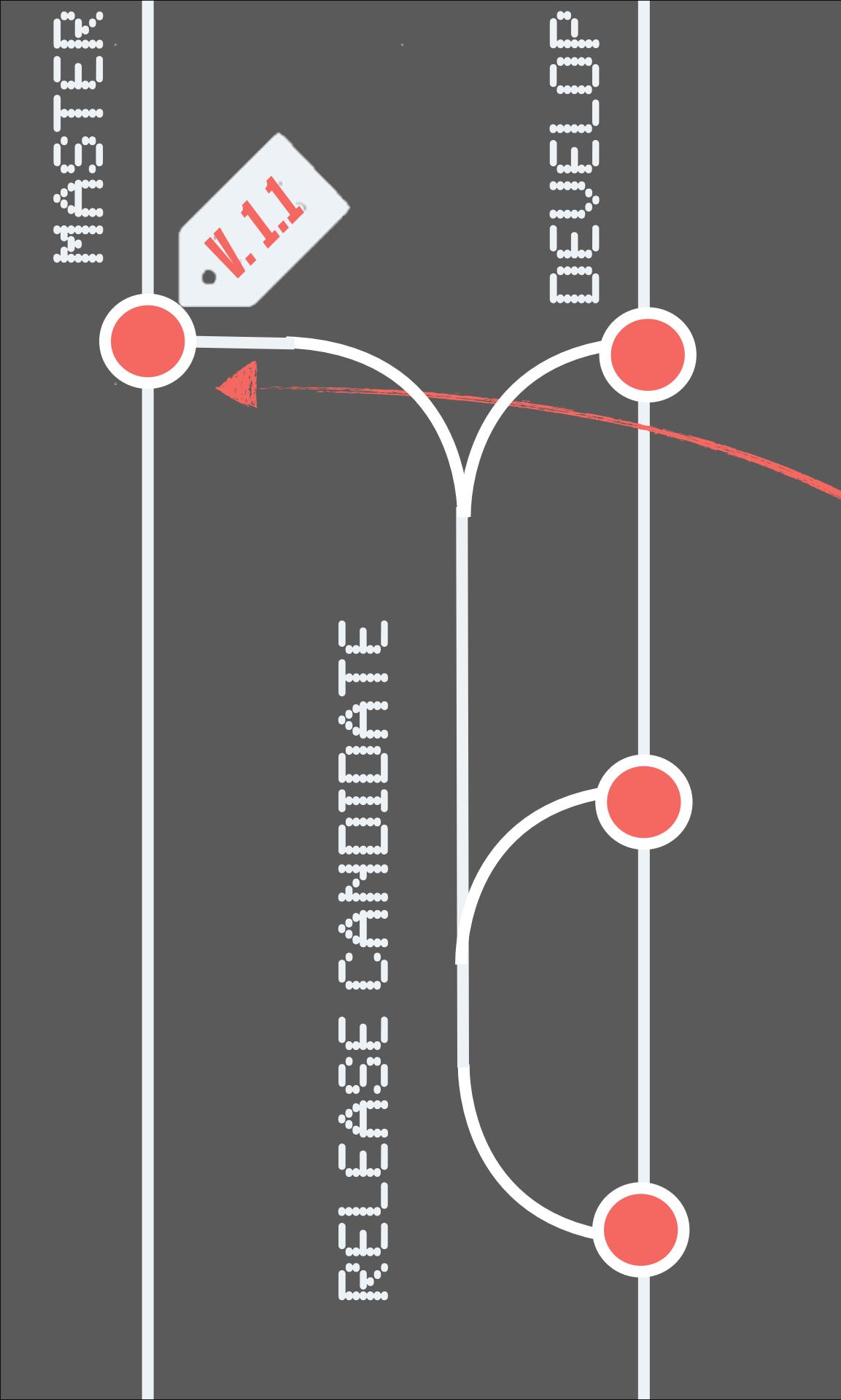
RELEASE CANDIDATE

DEVELOP



RELEASE

fixing back to DEVELOP



RELEASE

merge, tag, release!

**RELEASES ARE
EXPENSIVE**

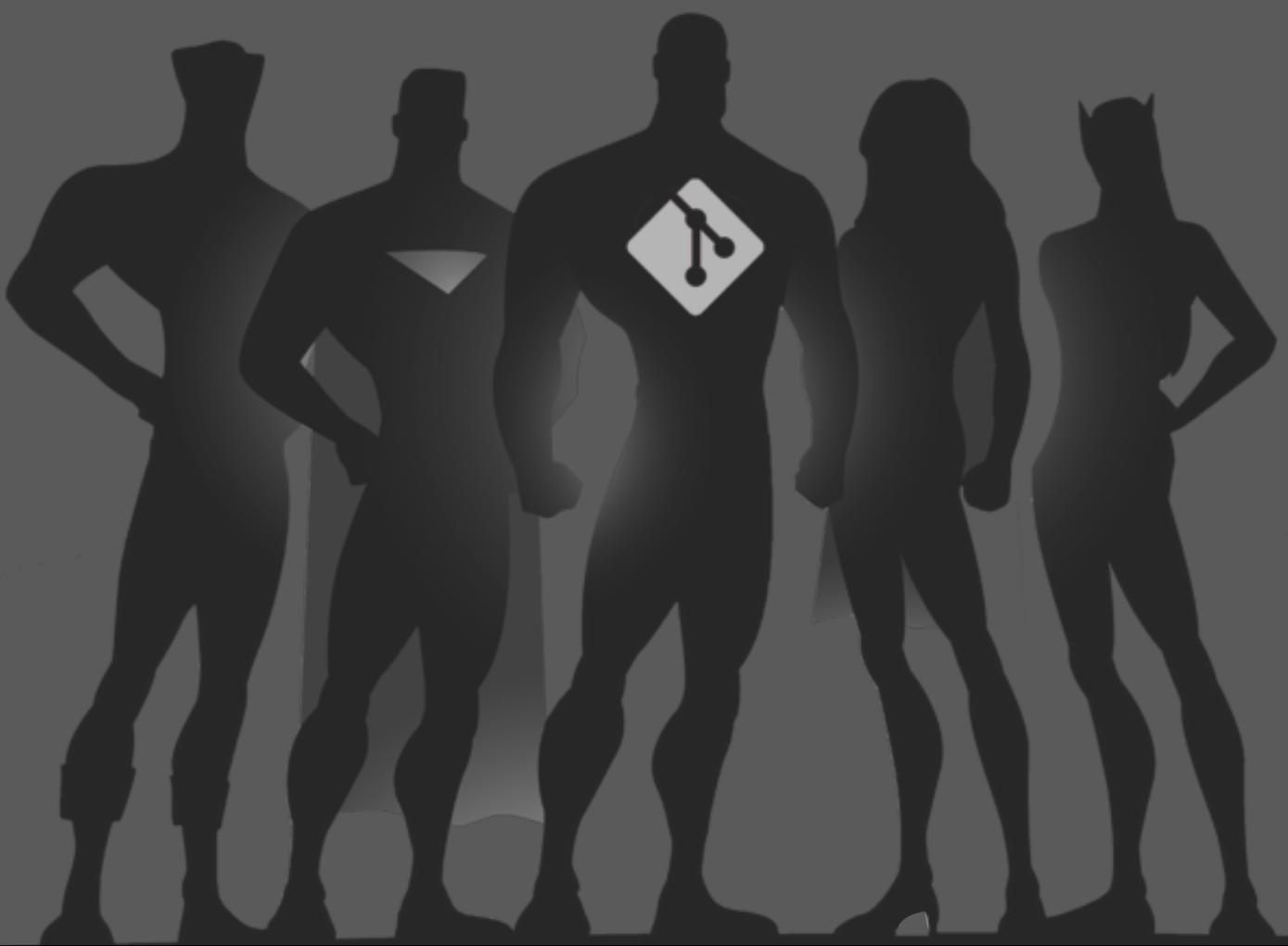
**BUGS HAVE A BIG
IMPACT**

**QUALITY OVER
SPEED**



**AFRAID OF
FAILURES**

THE MAINTENANCE WORKFLOW

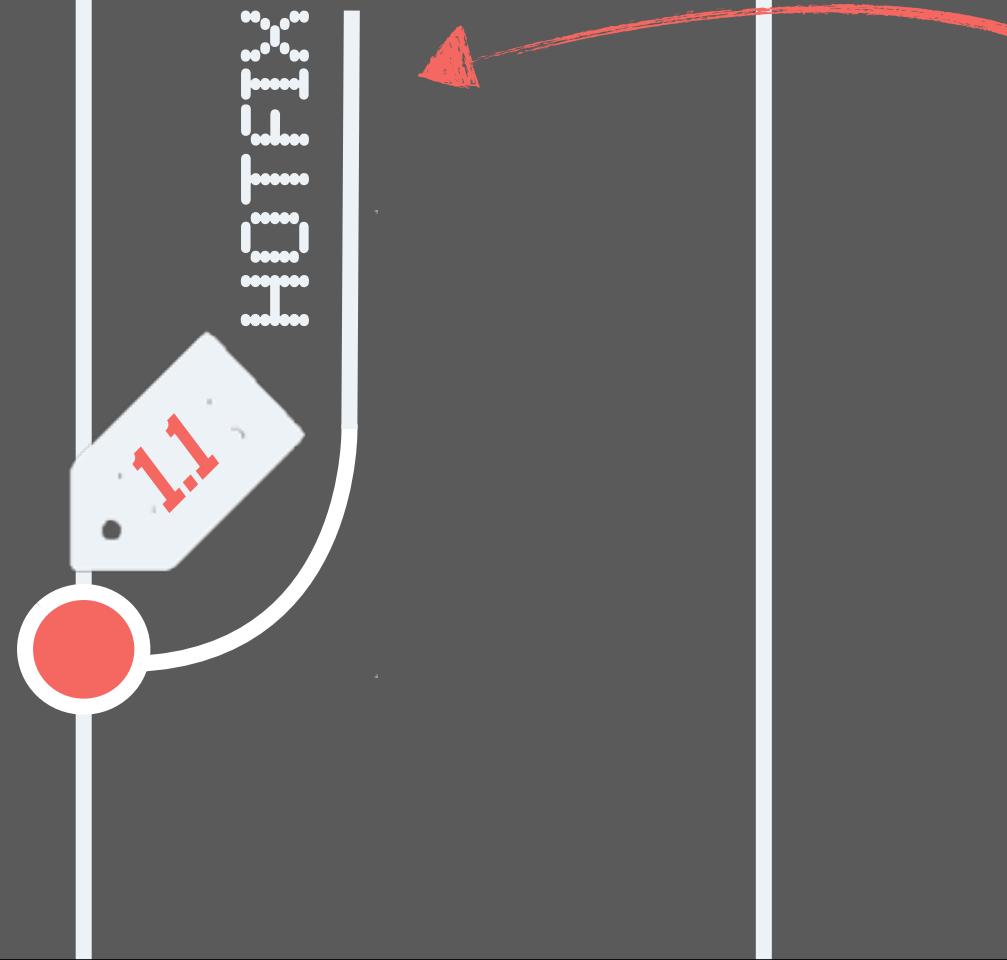


RELEASED SOFTWARE HAS BUGS

Houston we have a problem

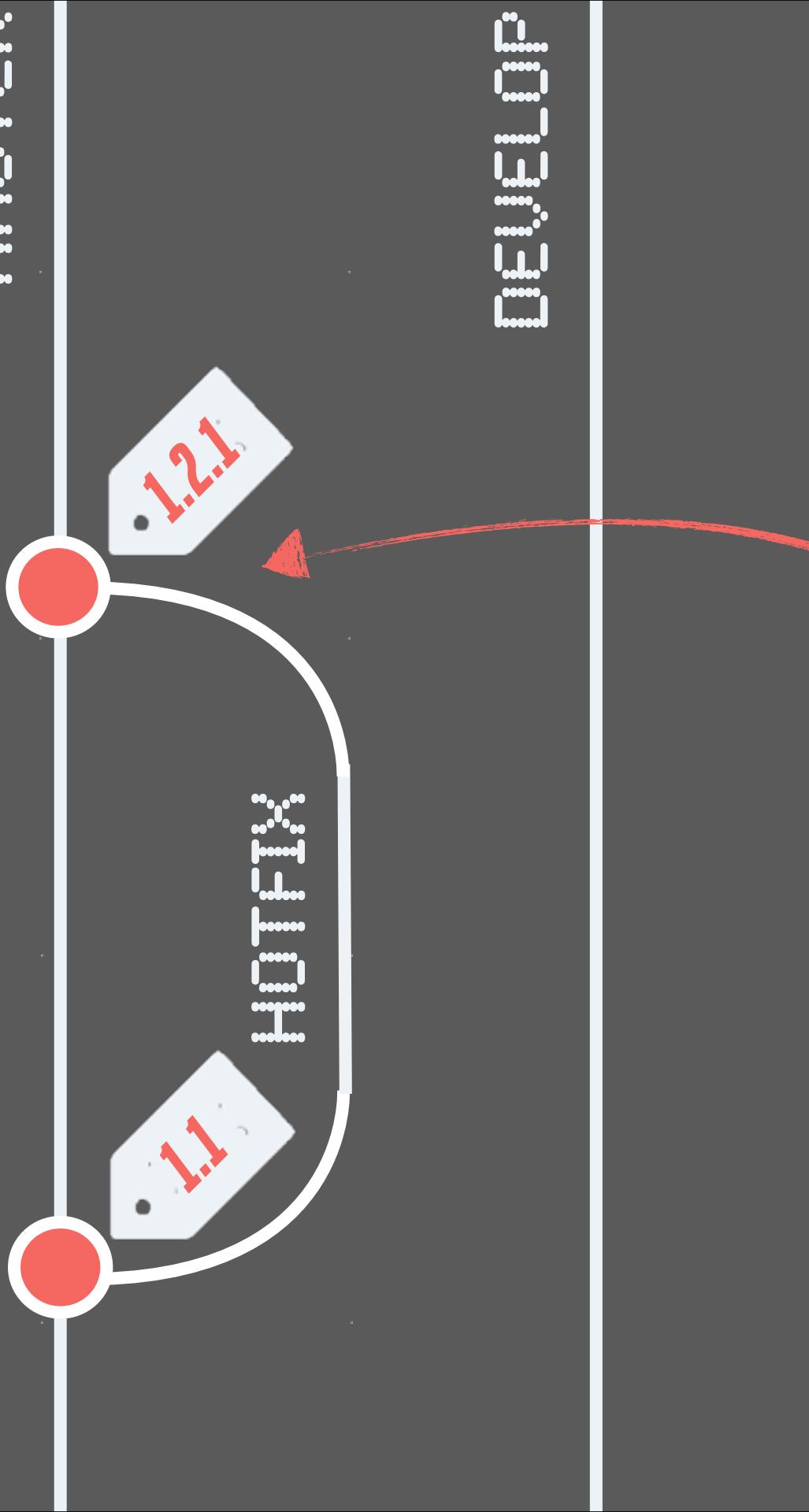


POS 3014



branch from MASTER and fix
the damn bug

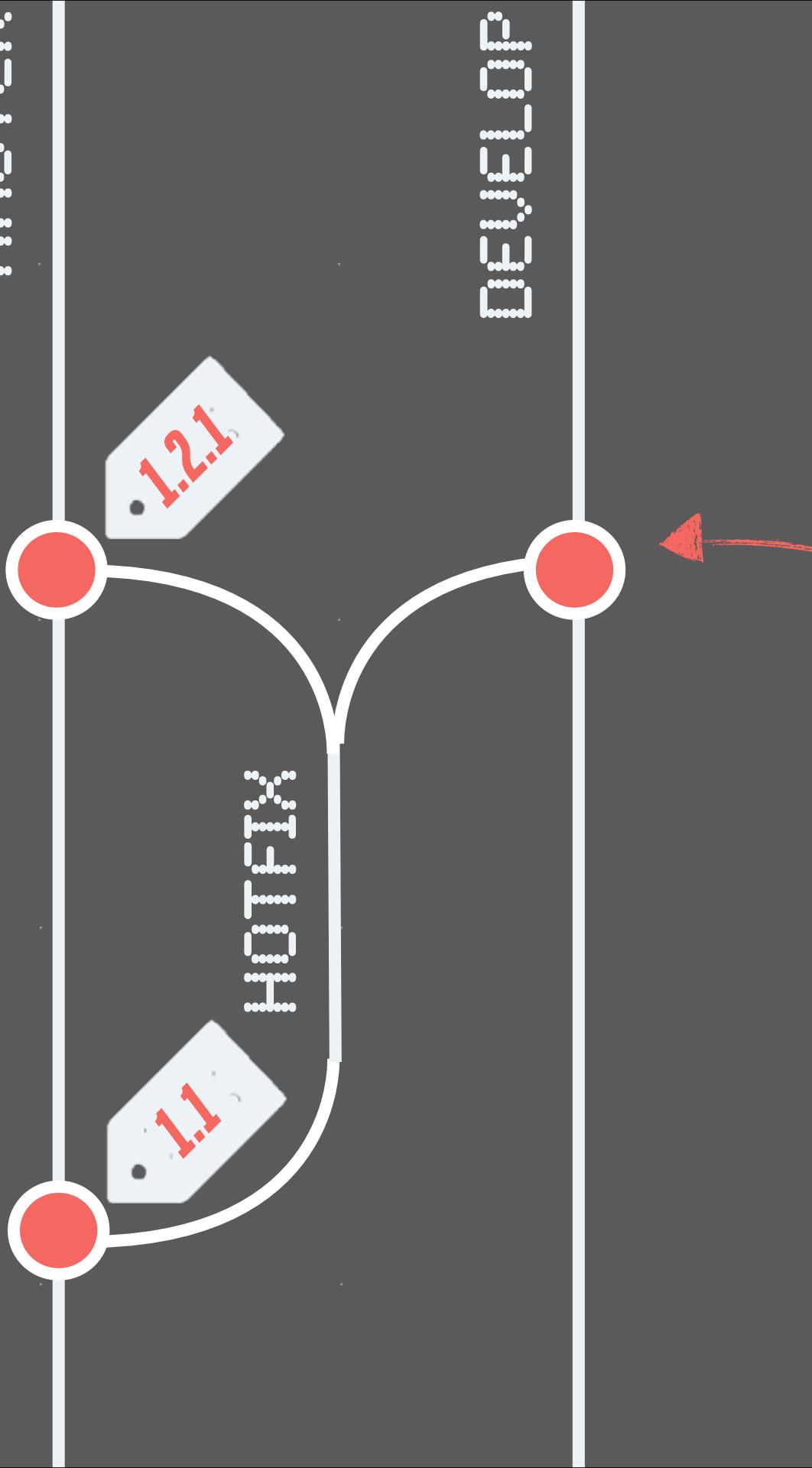
MAINTENANCE



MAINTENANCE

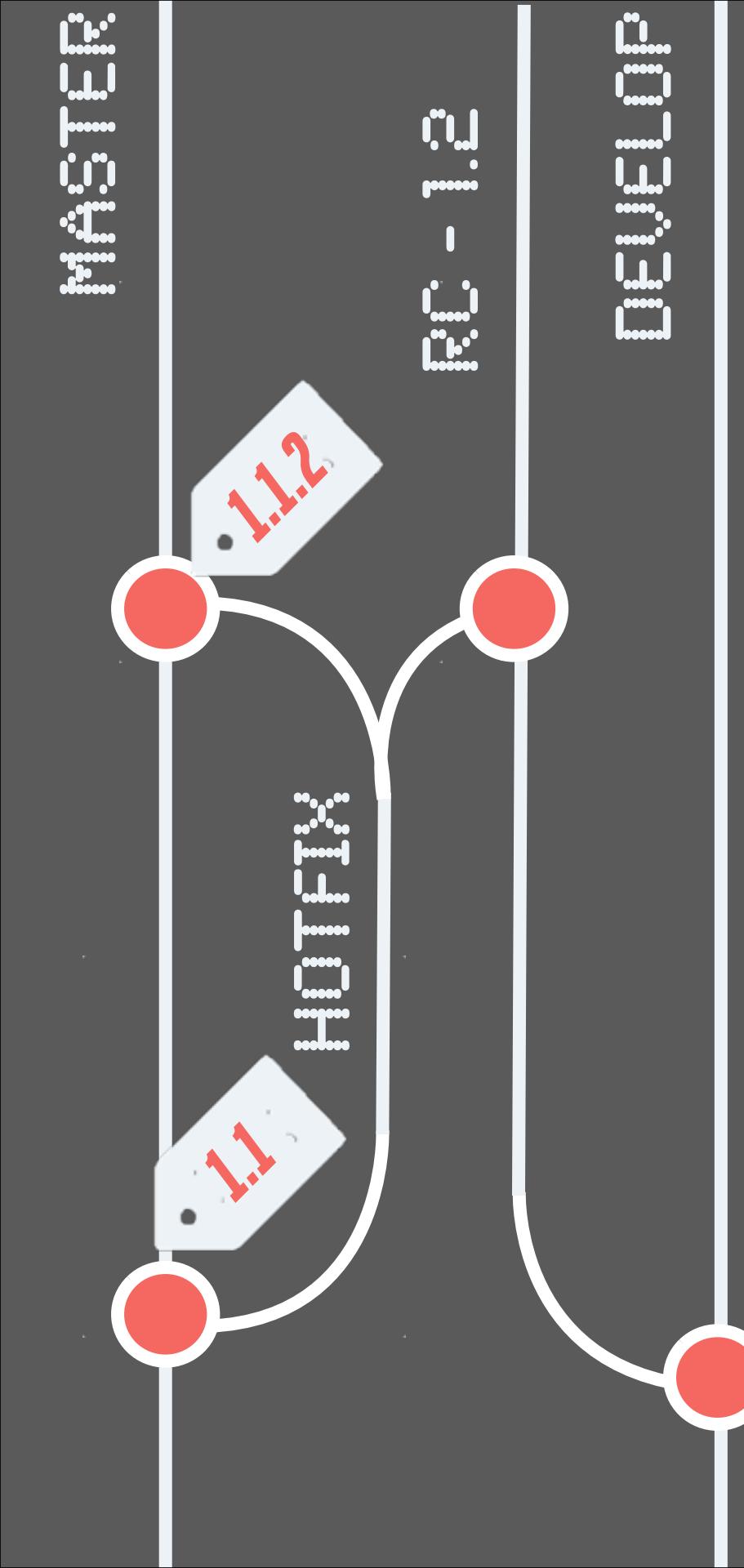
merge with MASTER, tag and
release

BEST PRACTICE



MAINTENANCE

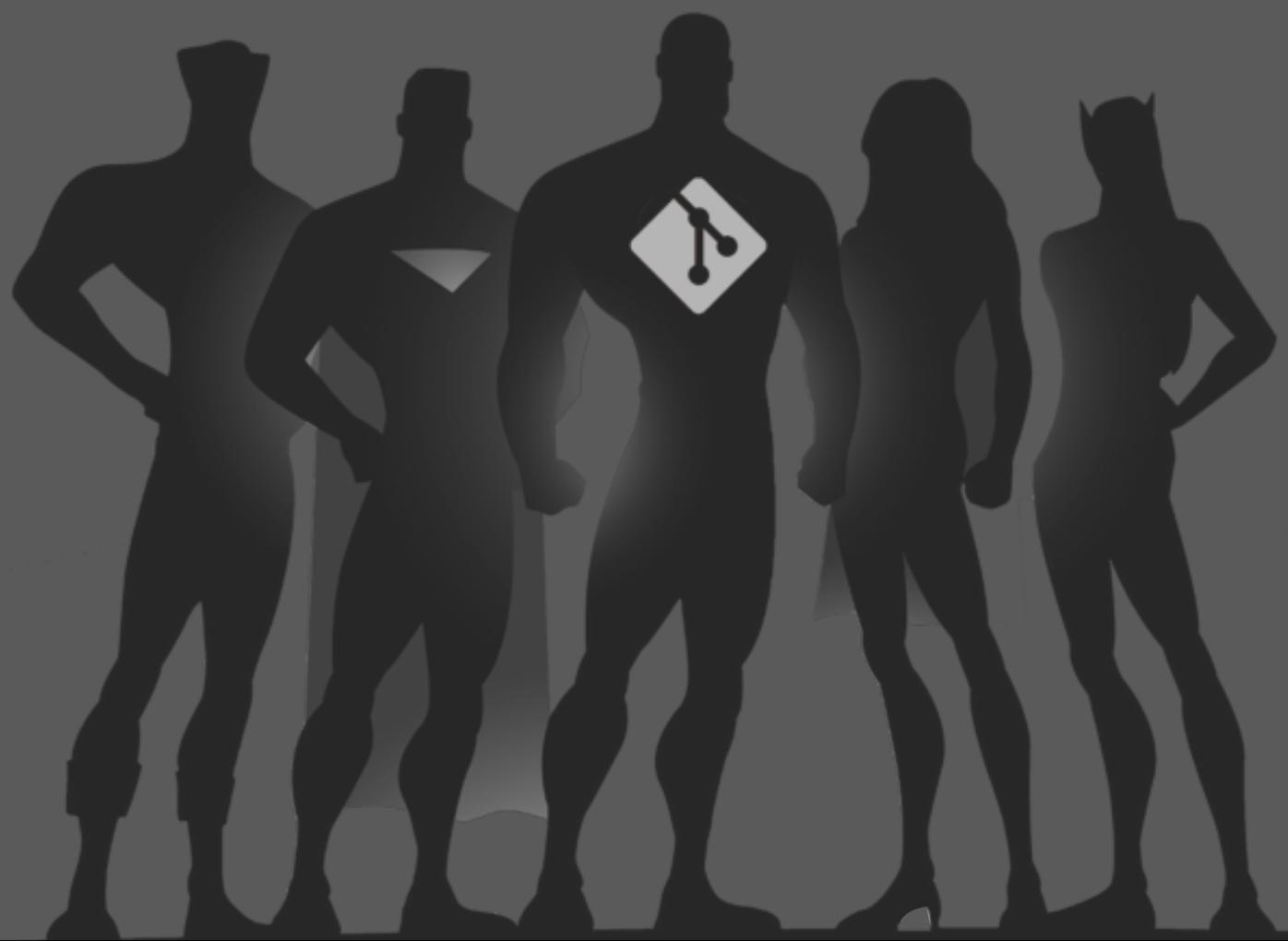
merge with DEVELOP and
delete HOTFIX



MAINTENANCE

ONE EXCEPTION:
A RELEASE BRANCH EXISTS

THE GITFLOW



YOU'VE JUST SEEN IT!

THE GITFLOW

DEVELOP & MASTER BRANCH

FEATURE BRANCHES

RELEASE BRANCHES

MAINTENANCE BRANCHES



from **Vincent Driessens**

THE GITFLOW NAMING CONVENTION

DEVELOP & MASTER BRANCH

FEATURE BRANCHES

RELEASE BRANCHES

MAINTENANCE BRANCHES

THE GITFLOW NAMING CONVENTION

develop & master

FEATURE BRANCHES

RELEASE BRANCHES

MAINTENANCE BRANCHES

THE GITFLOW NAMING CONVENTION

develop & master

feature-*

RELEASE BRANCHES

MAINTENANCE BRANCHES

THE GITFLOW NAMING CONVENTION

develop & master

feature-*

release-*

MAINTENANCE BRANCHES

THE GITFLOW NAMING CONVENTION

develop & master

feature-*

release-*

hotfix-*

Or create your own!

HOW CAN I FIND MY WAY THROUGH THE RULE JUNGLE?



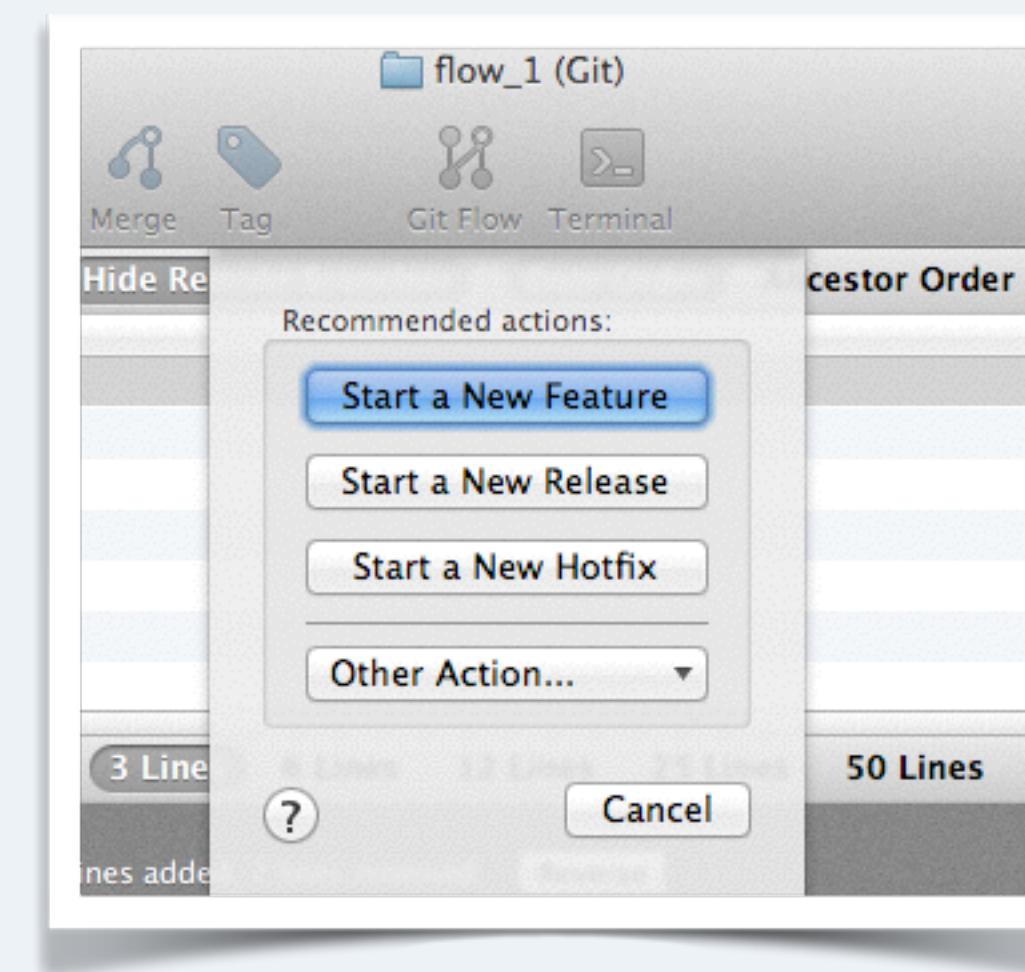
HOW CAN I REMEMBER ALL THESE *GIT COMMANDS AND RULES?*

```
git flow feature start 'JRA-34 Add User'  
git flow feature finish 'JRA-34 Add User'  
git flow release start 1.6  
git flow hotfix finish 1.5.3
```



there is a command line tool

HOW CAN I REMEMBER ALL THESE GIT COMMANDS AND RULES?



Atlassian
SourceTree

WHY **GITFLOW**



► I thought every team is different?

WHY **GITFLOW**

IT CAPTURES A LOT OF DEVELOPMENT SITUATIONS

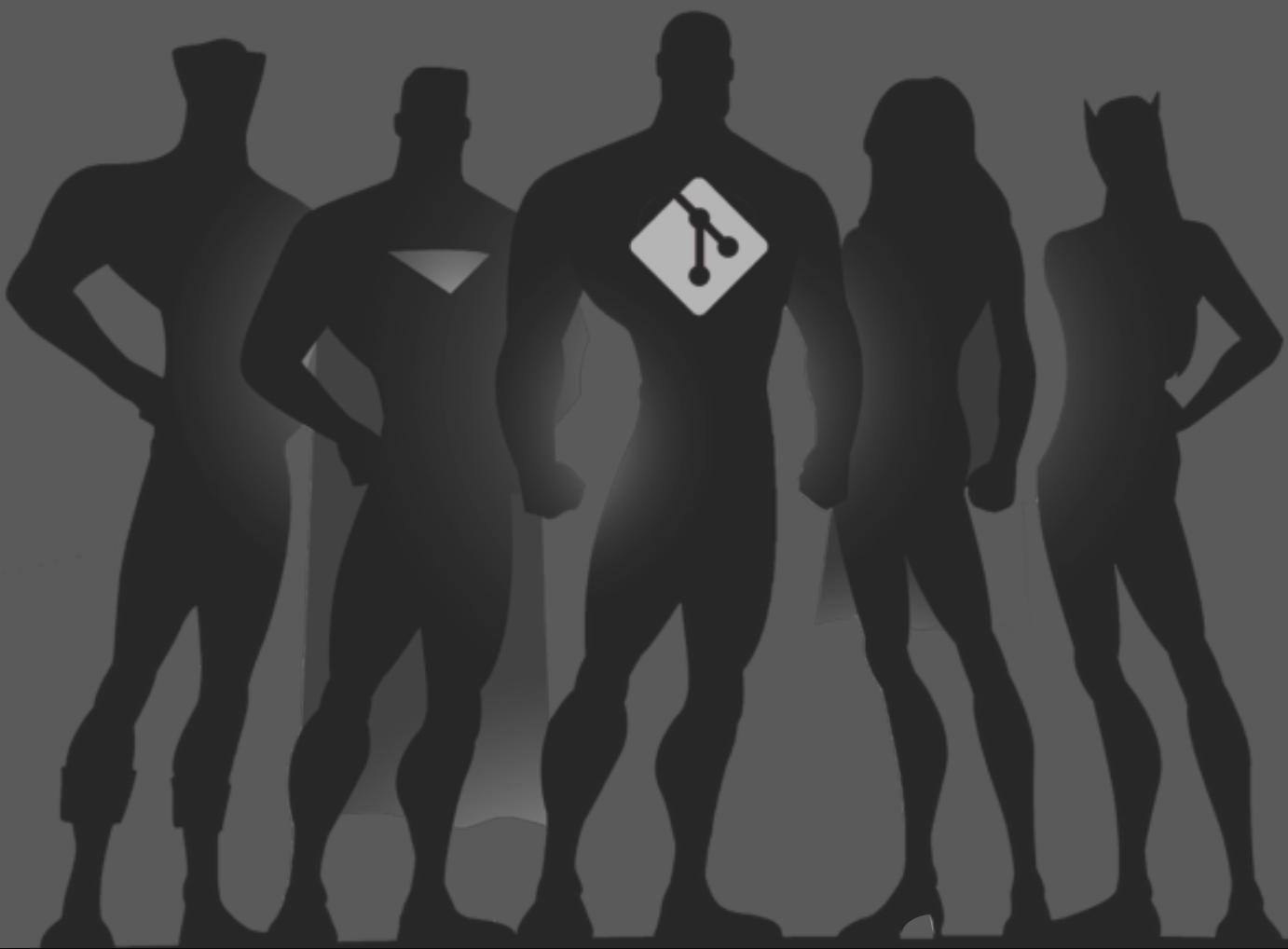
IT'S WELL KNOWN: EASY ADOPTION

BUT

I WON'T USE IT OUT-OF-THE-BOX: ADAPT IT!

THE STASH TEAM FLOW

at Atlassian



MAKING *GITFLOW* MORE SIMPLE



getting rid of MASTER

KEEP YOUR



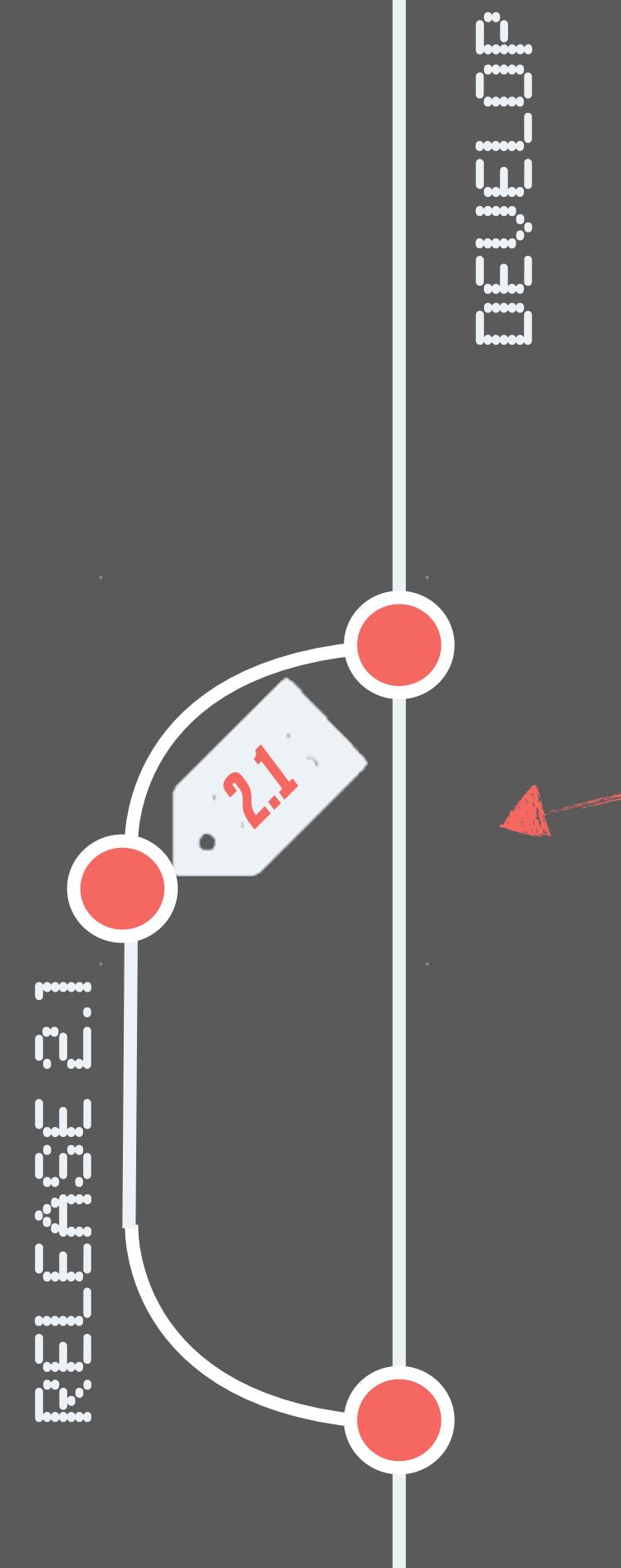
RELEASE



BRANCH



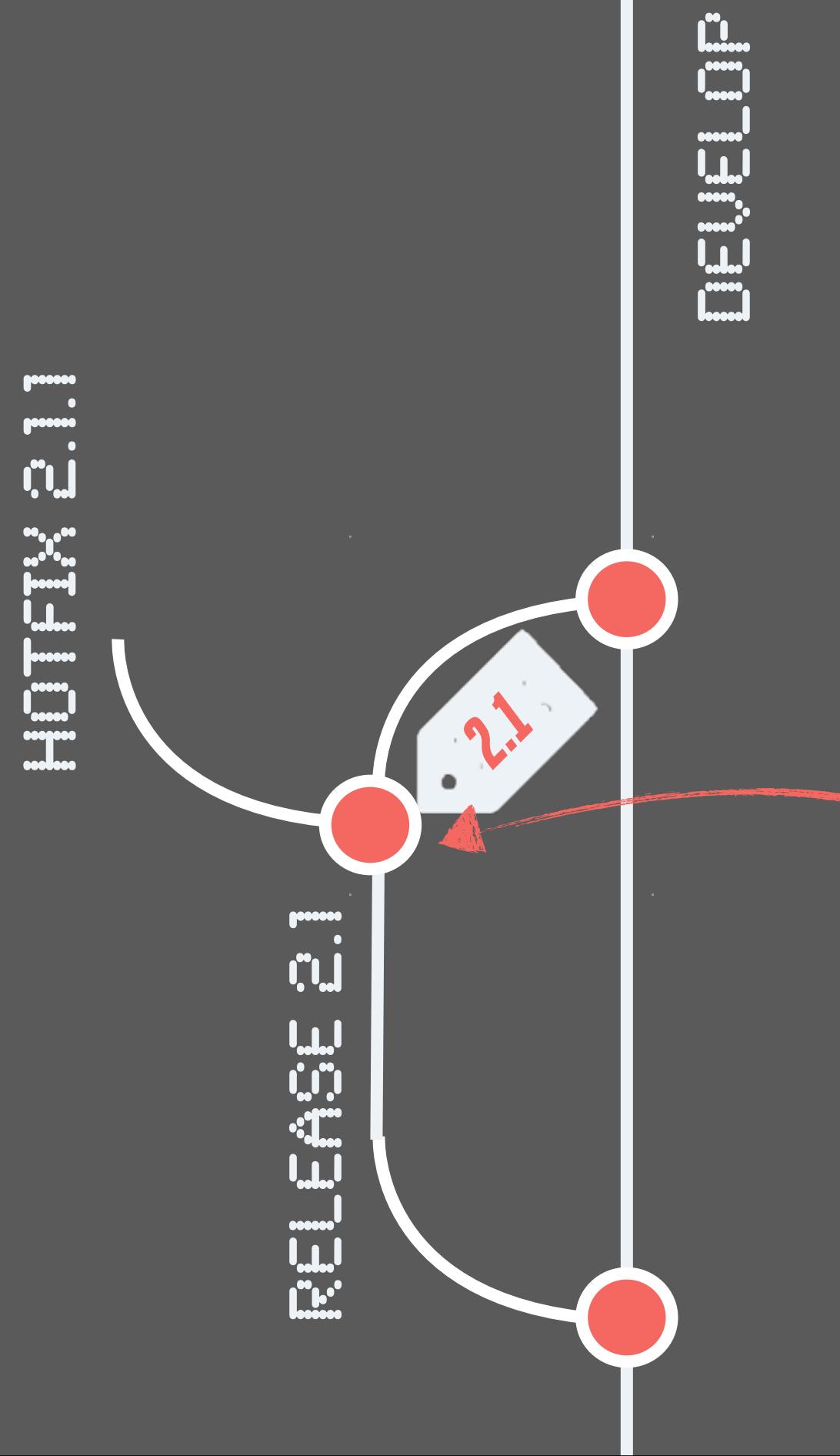
no merge into MASTER needed



STASH TEAM FLOW

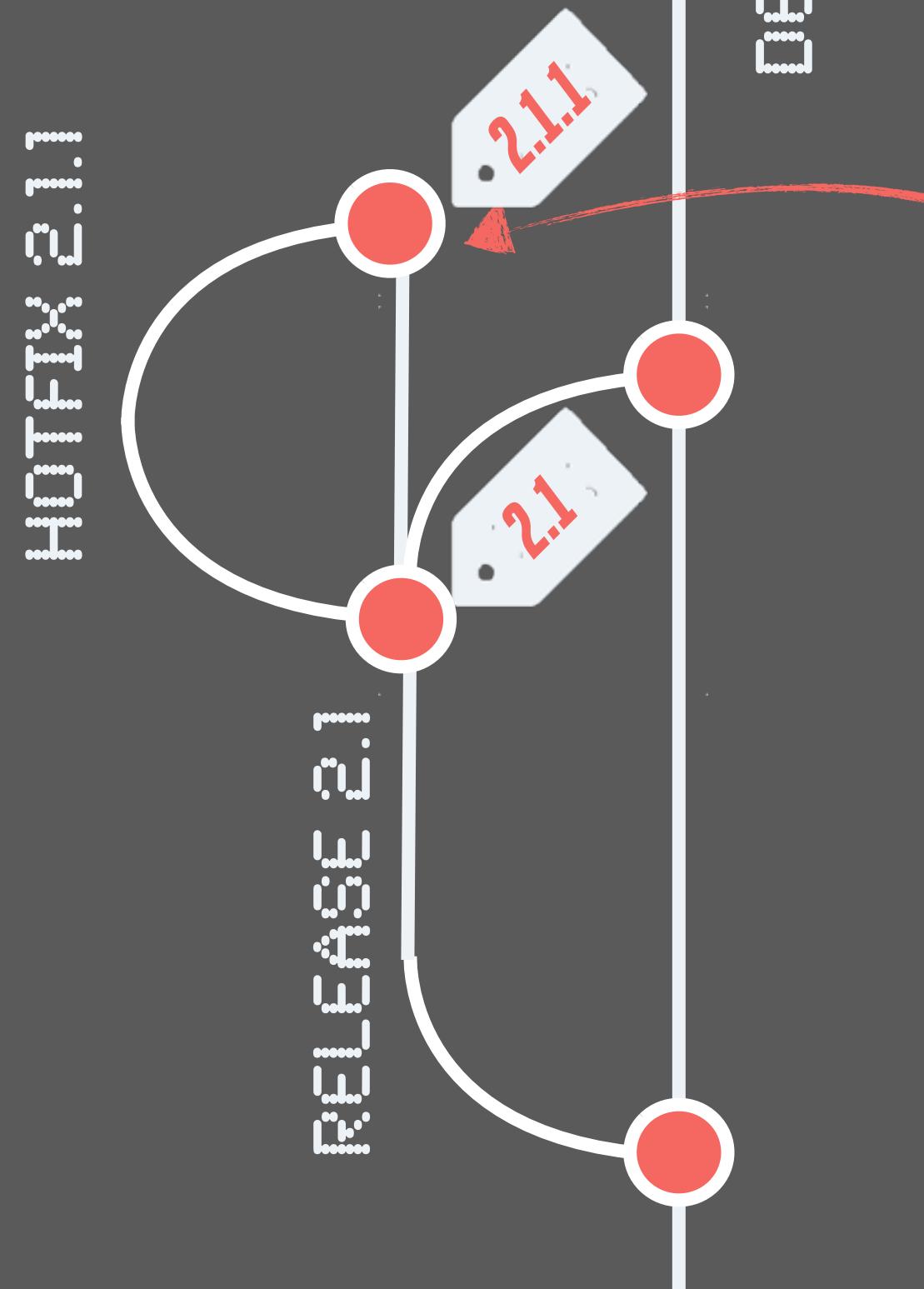
NO MASTER, we keep RELEASE

STASH TEAM FLOW



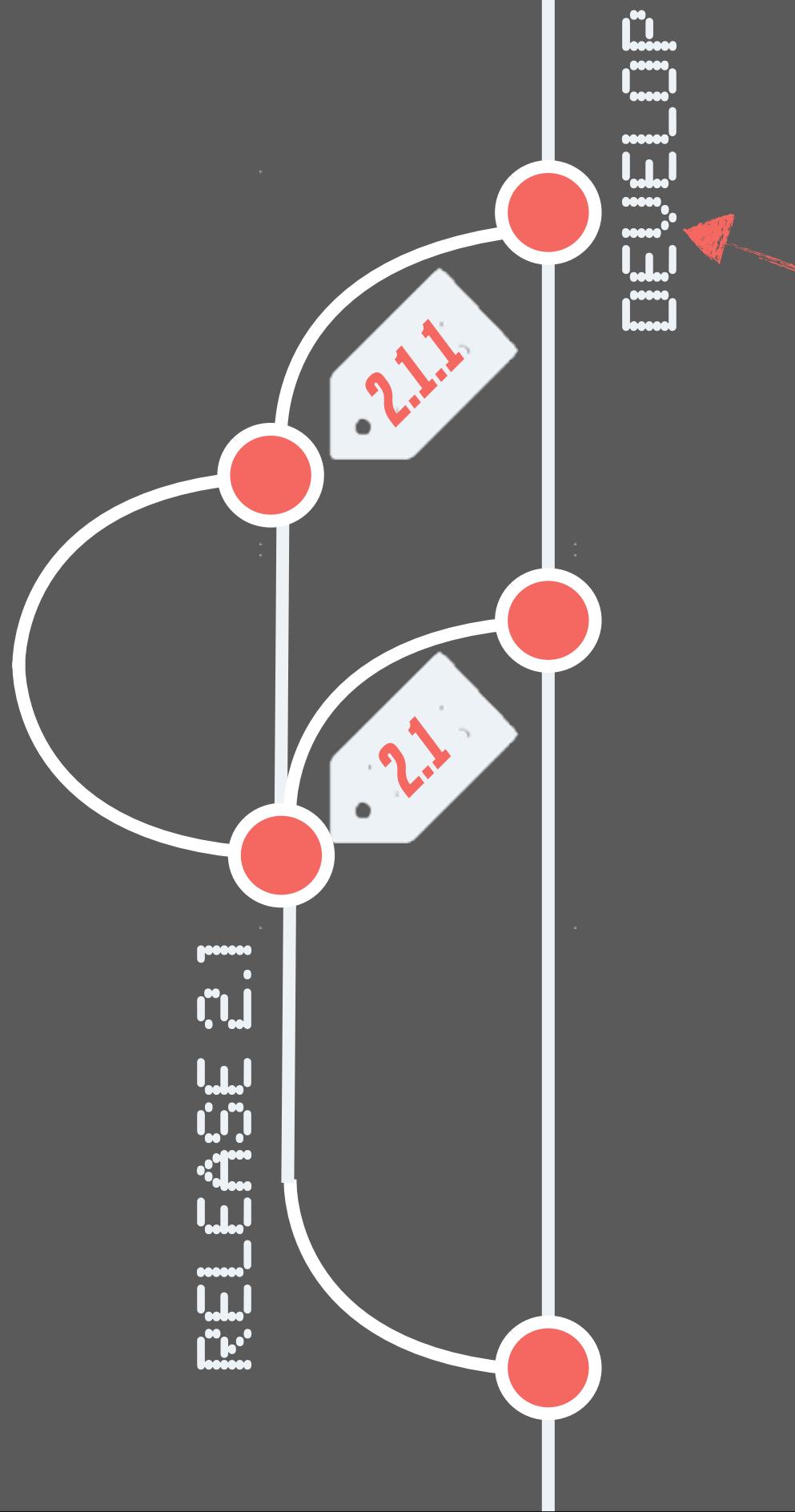
branch HOTFIX from RELEASE
and fix the damn bug

STASH TEAM FLOW



merge back to RELEASE

STASH TEAM FLOW



STASH TEAM FLOW

and apply changes to DEVELOP

they are still out there
...and have sometimes problems

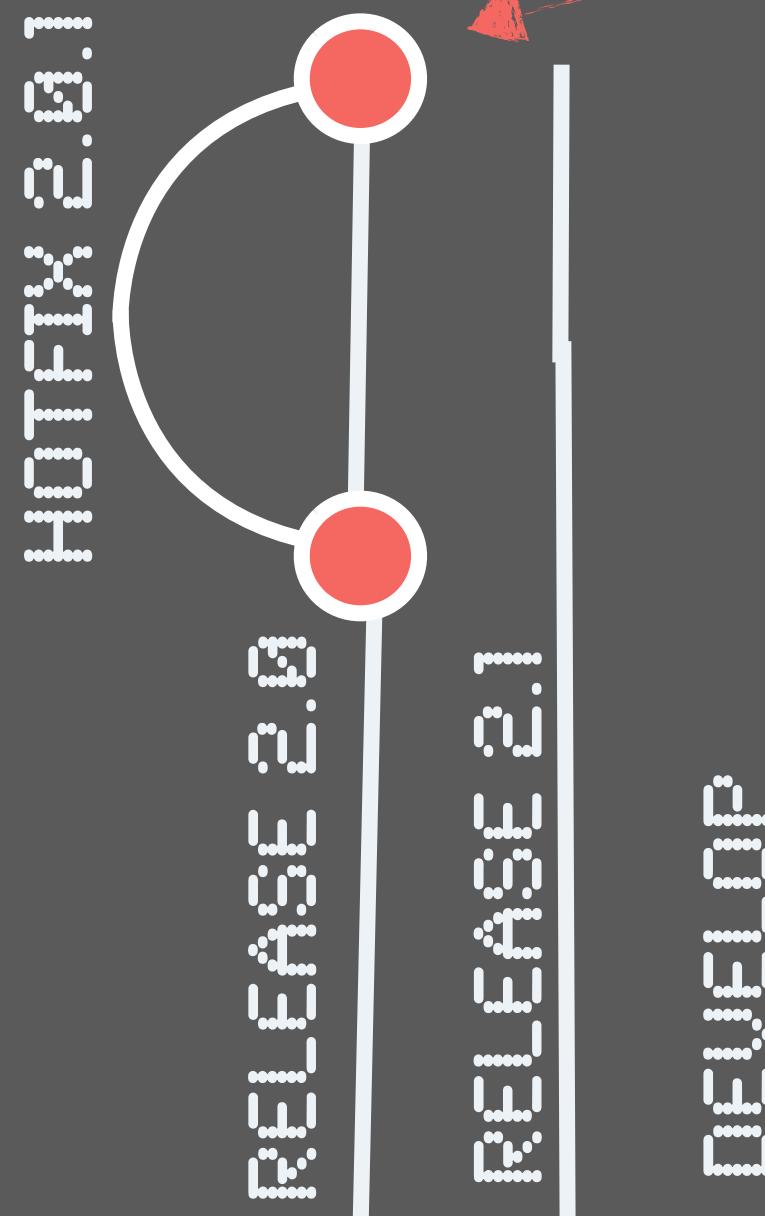


SUPPORTING OLD VERSIONS

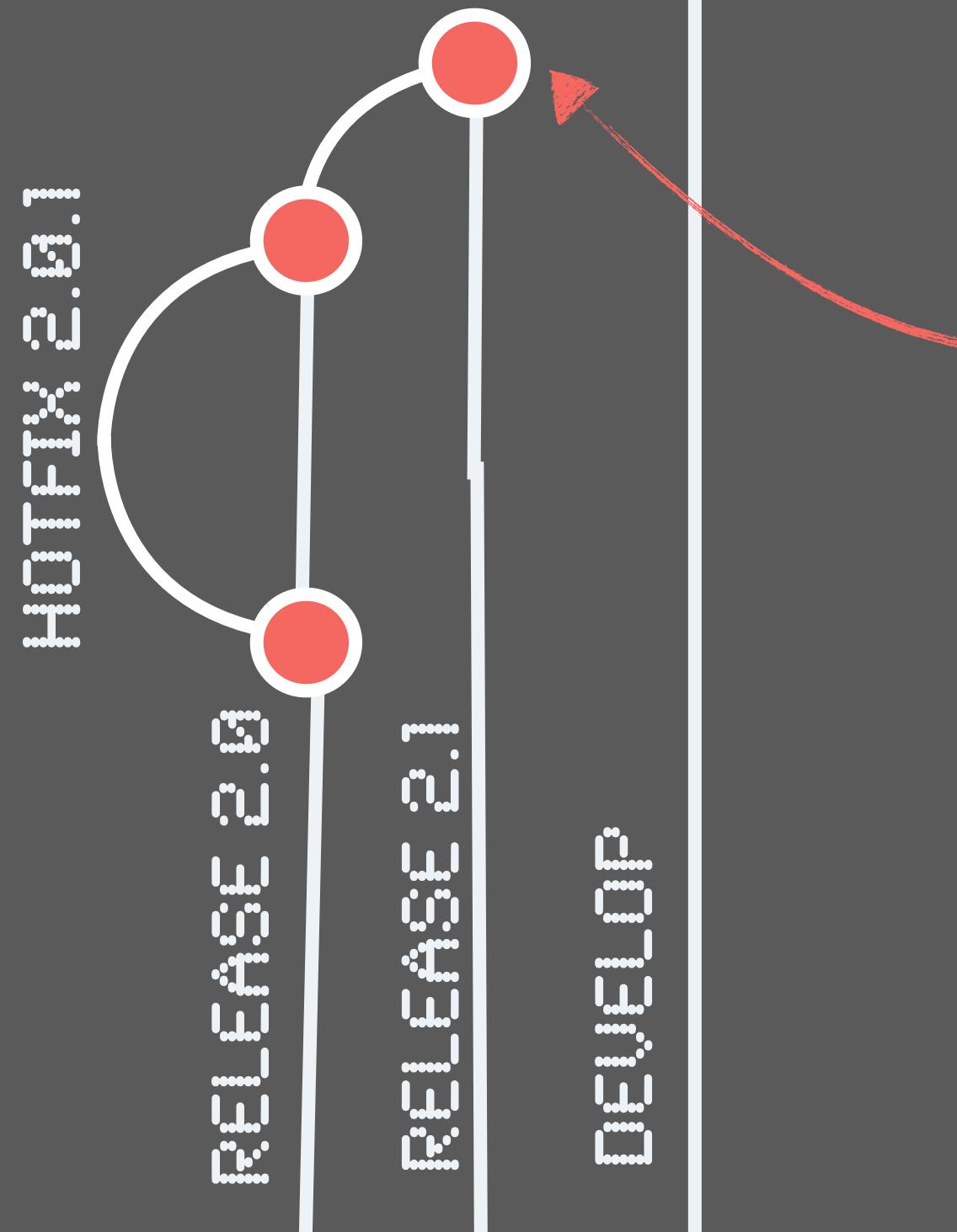
STASH TEAM FLOW



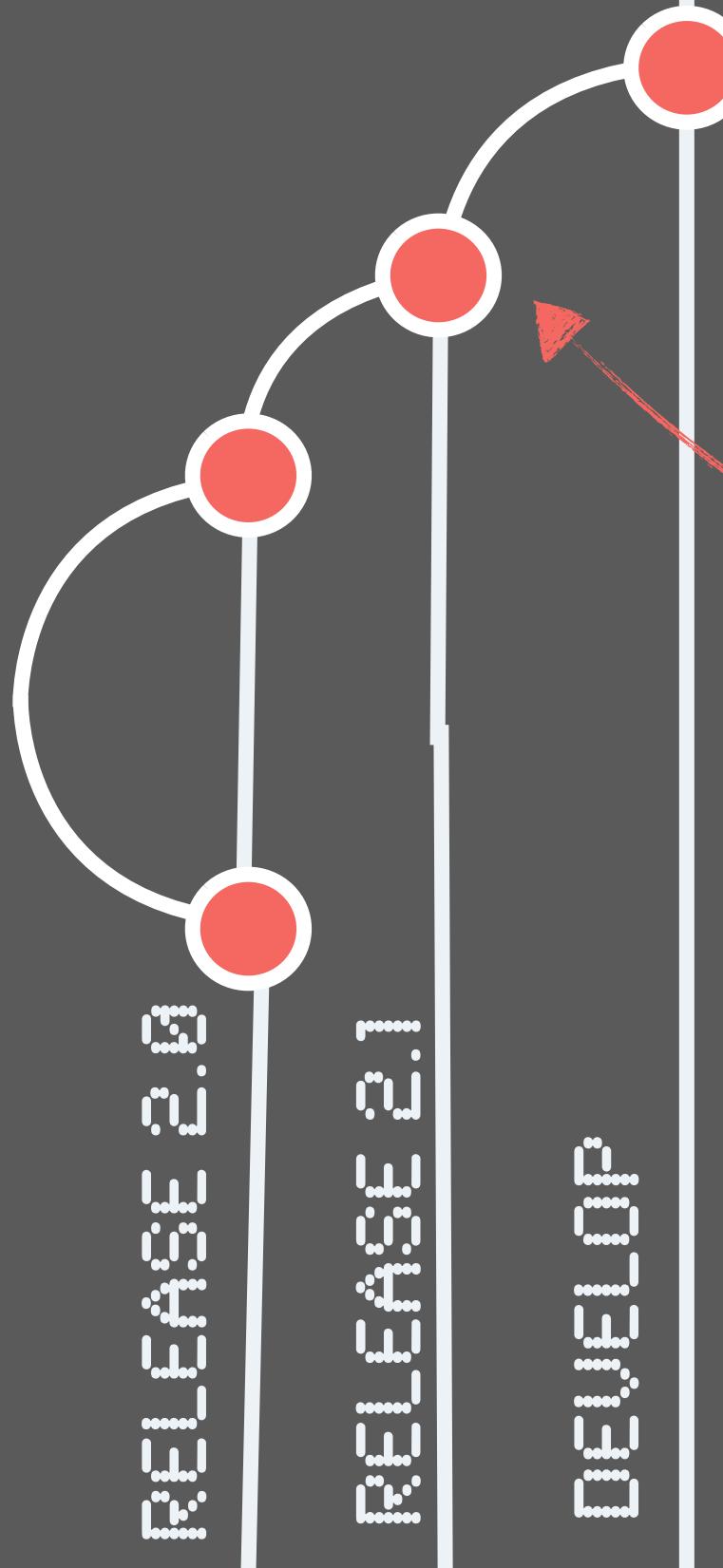
STASH TEAM FLOW



STASH TEAM FLOW



RELEASE 2.1



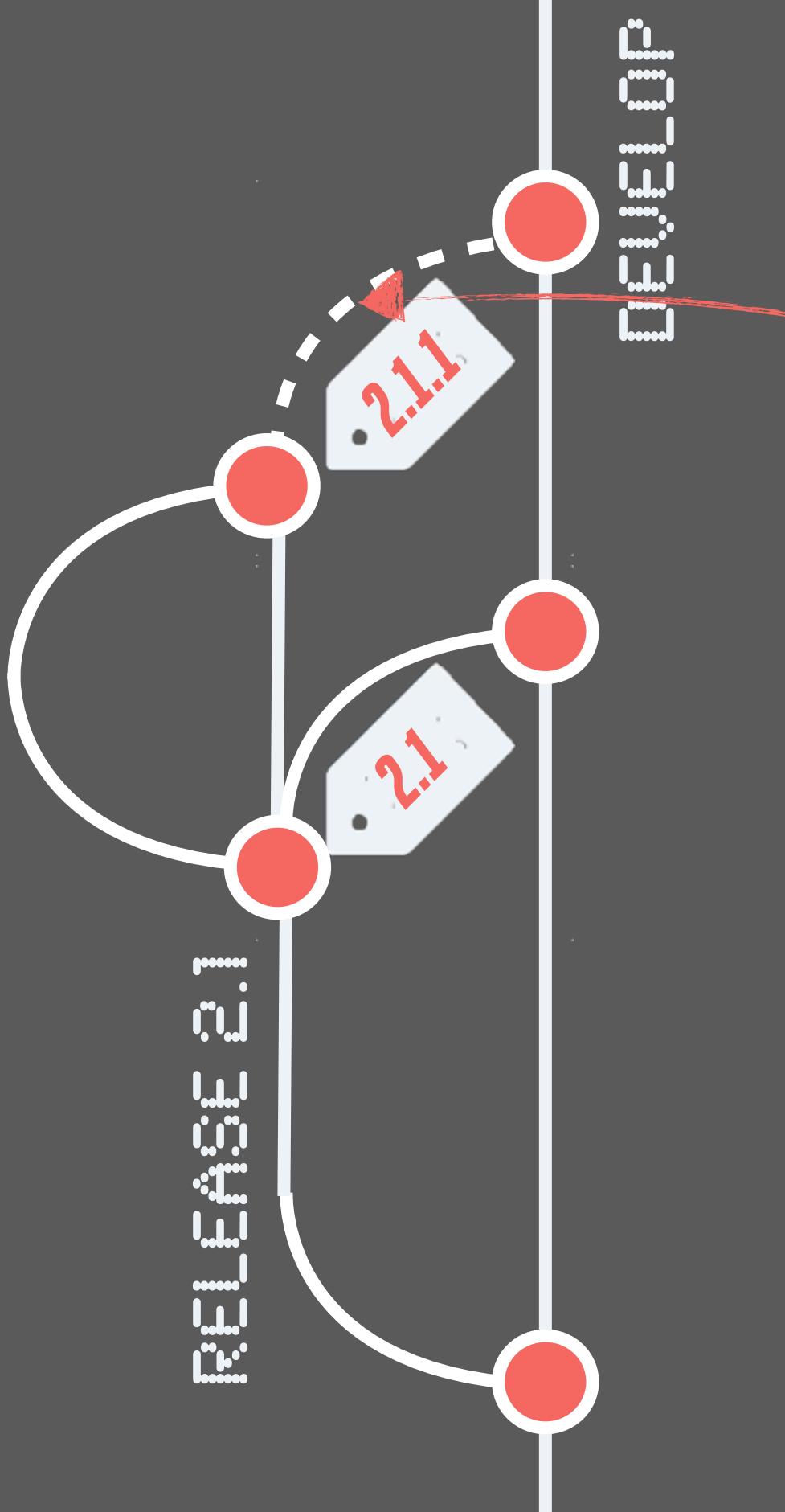
STASH TEAM FLOW

and apply changes from
RELEASE 2.1 to DEVELOP

STASH TEAM *FLOW*



We want to help



SERVER SIDE

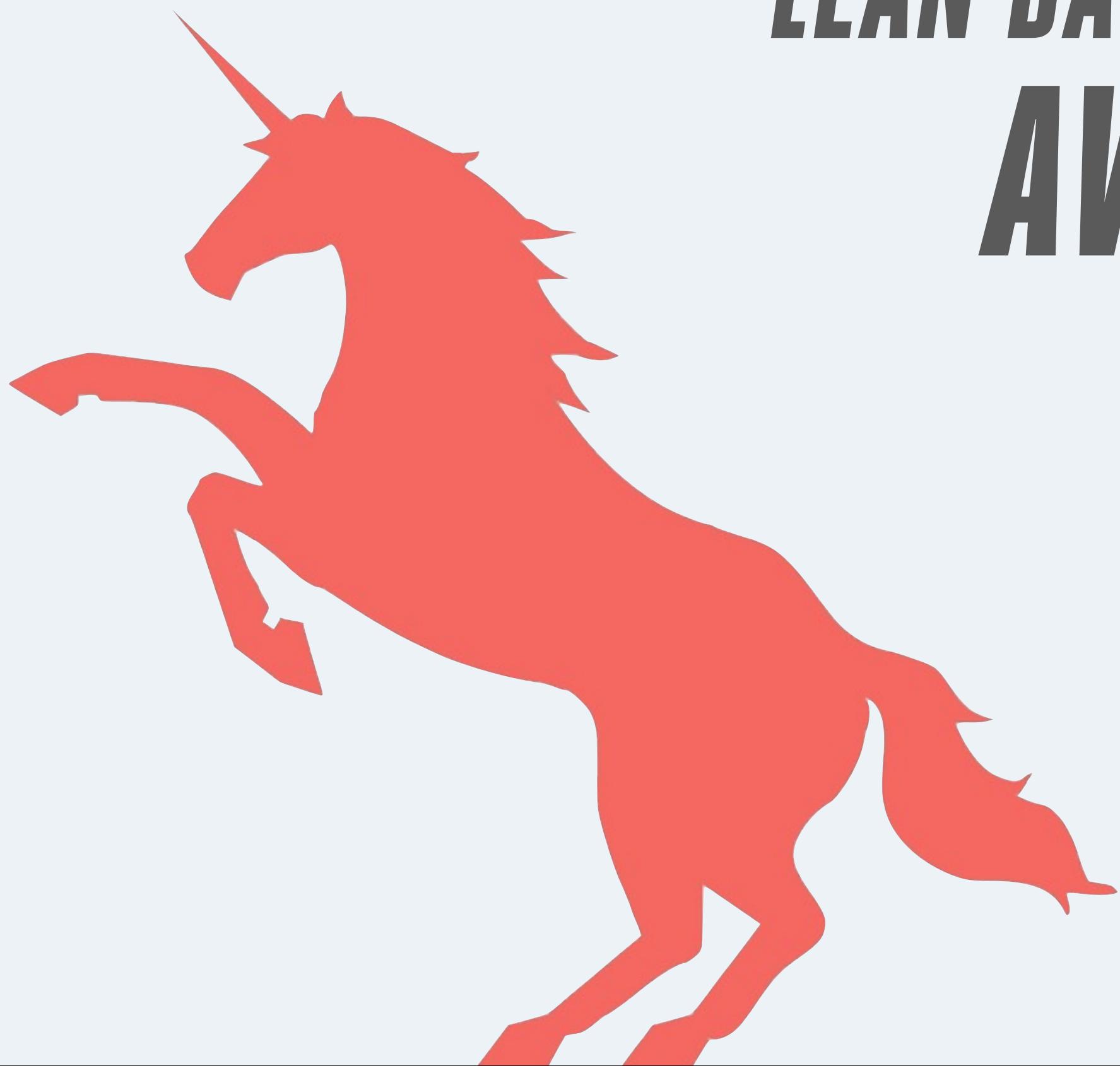
AUTO MERGES



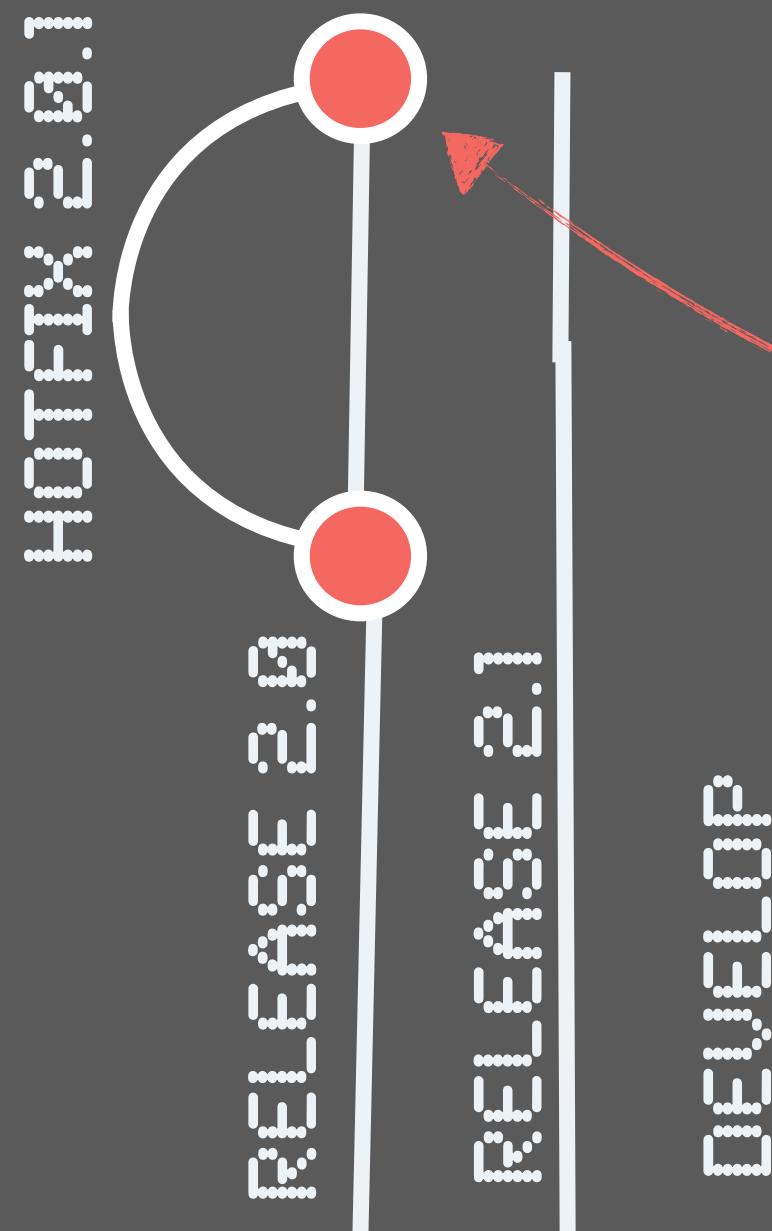
I could do this

with Git hooks

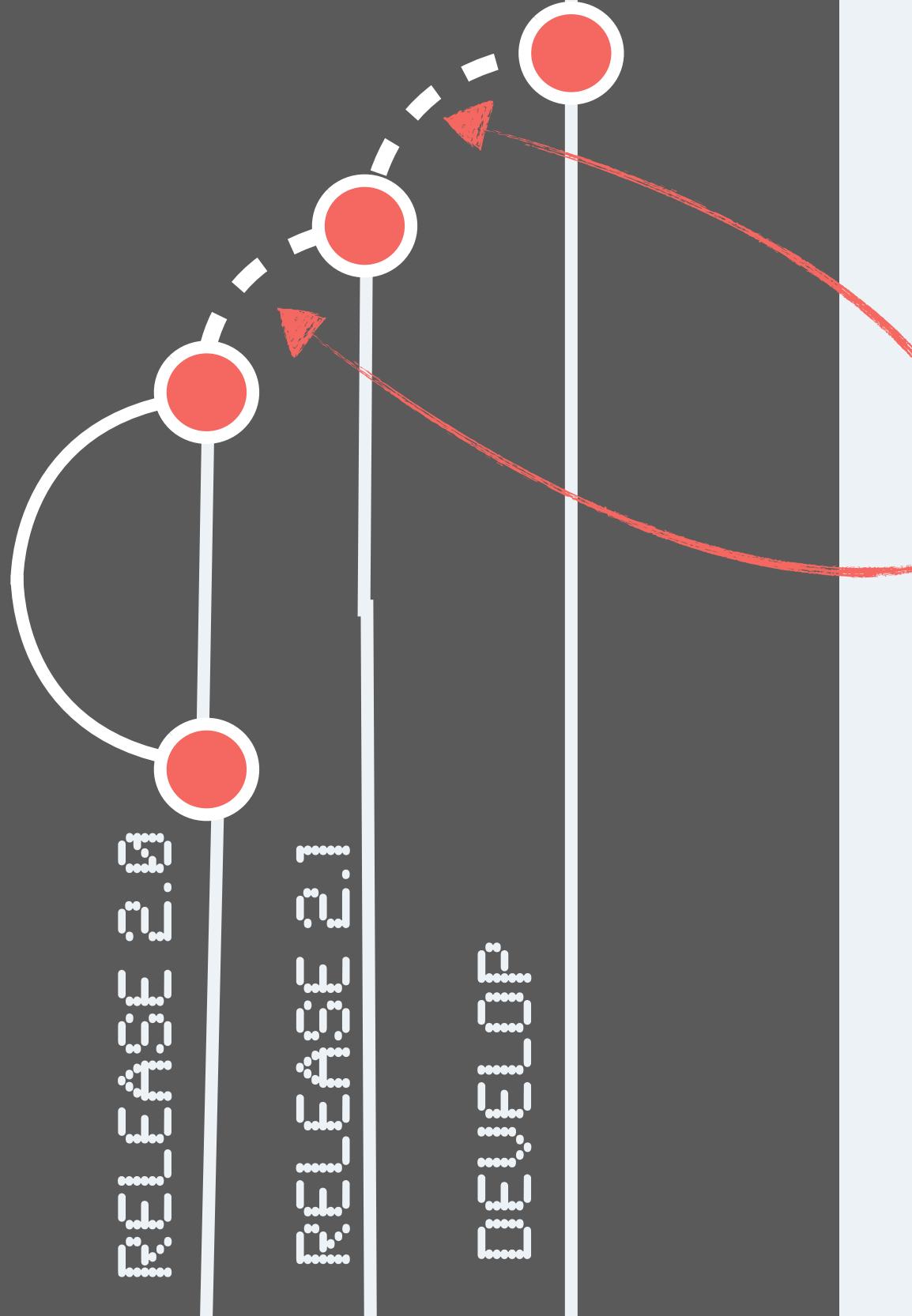
*LEAN BACK AND ENJOY PURE
AWESOMENESS*



STASH TEAM FLOW



DATA
SCIENCE

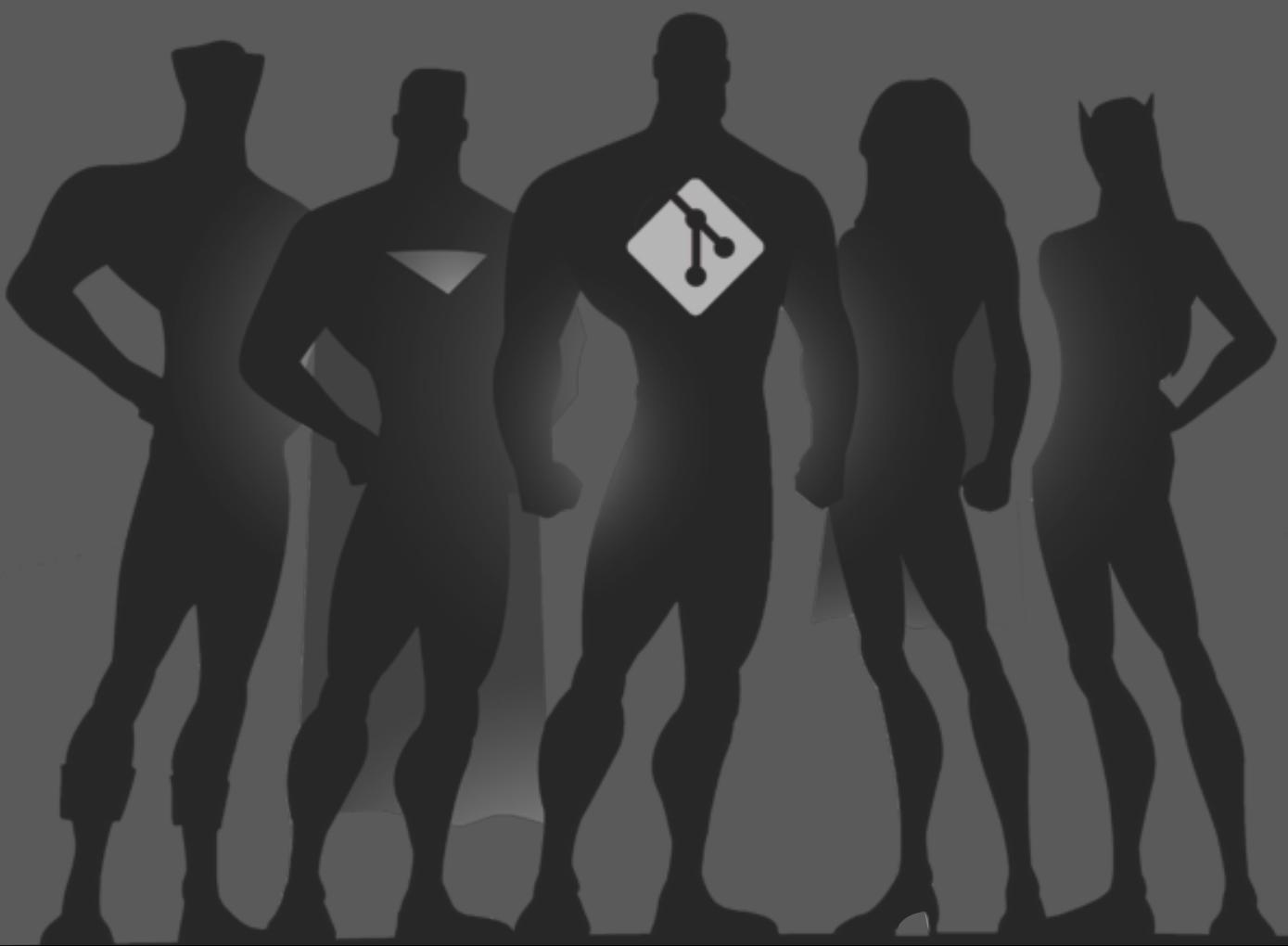


STASH TEAM FLOW

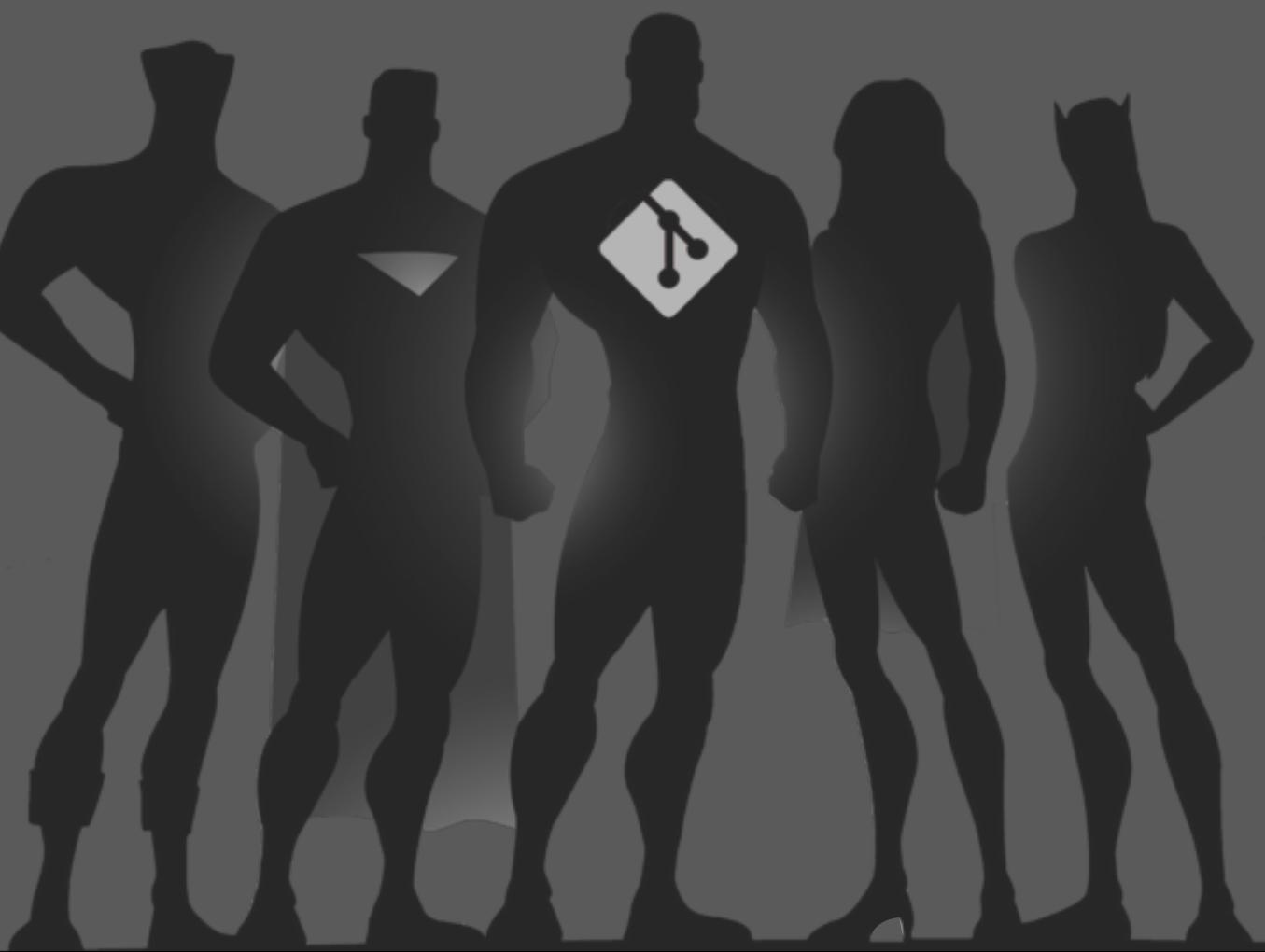
and let the robots take over



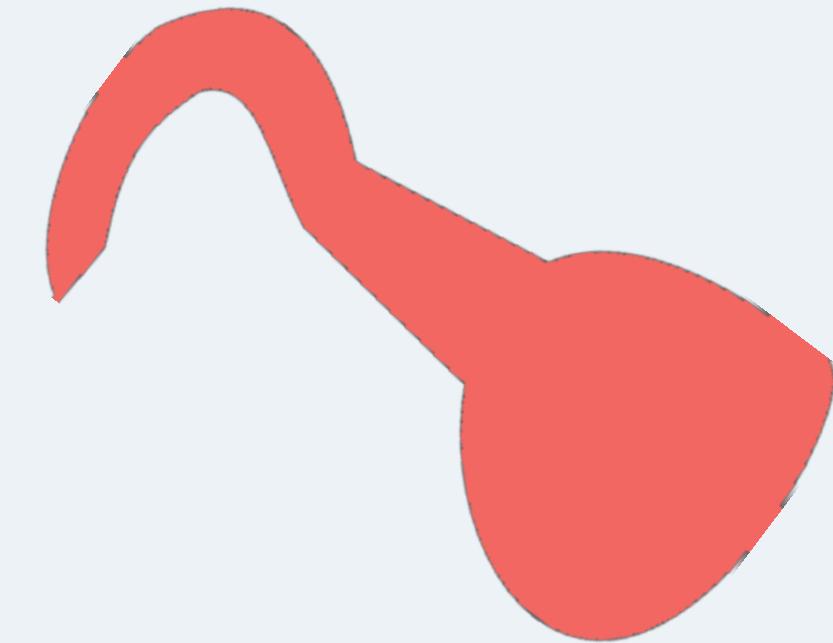
NOT ENOUGH POSSIBILITIES?



INFINITE WORKFLOW POSSIBILITIES



CUSTOMIZE YOUR WORKFLOW WITH *HOOKS*





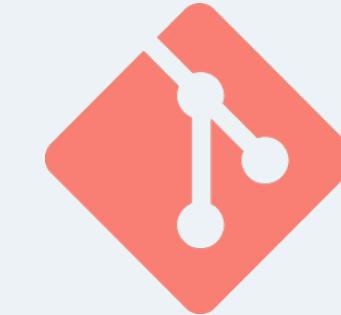
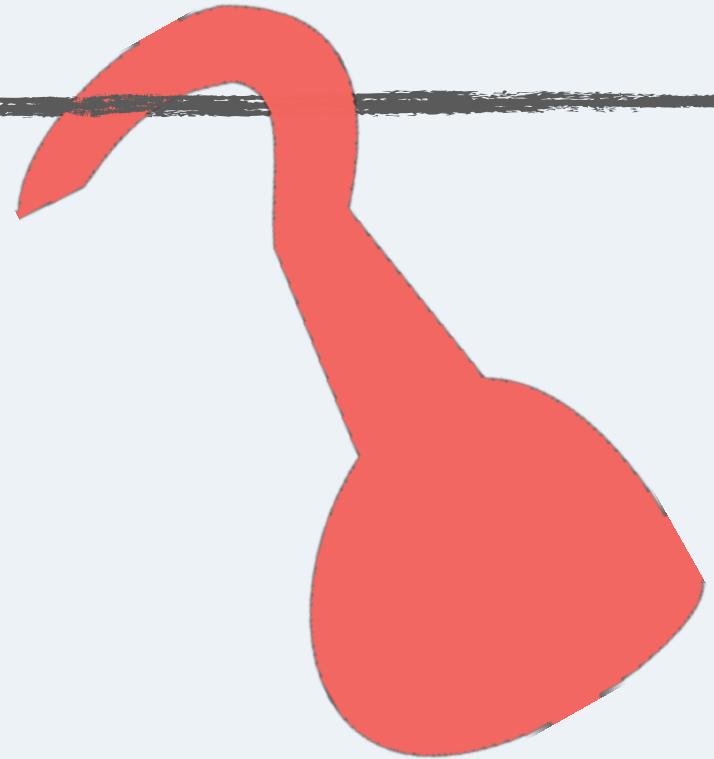
PUSH



git

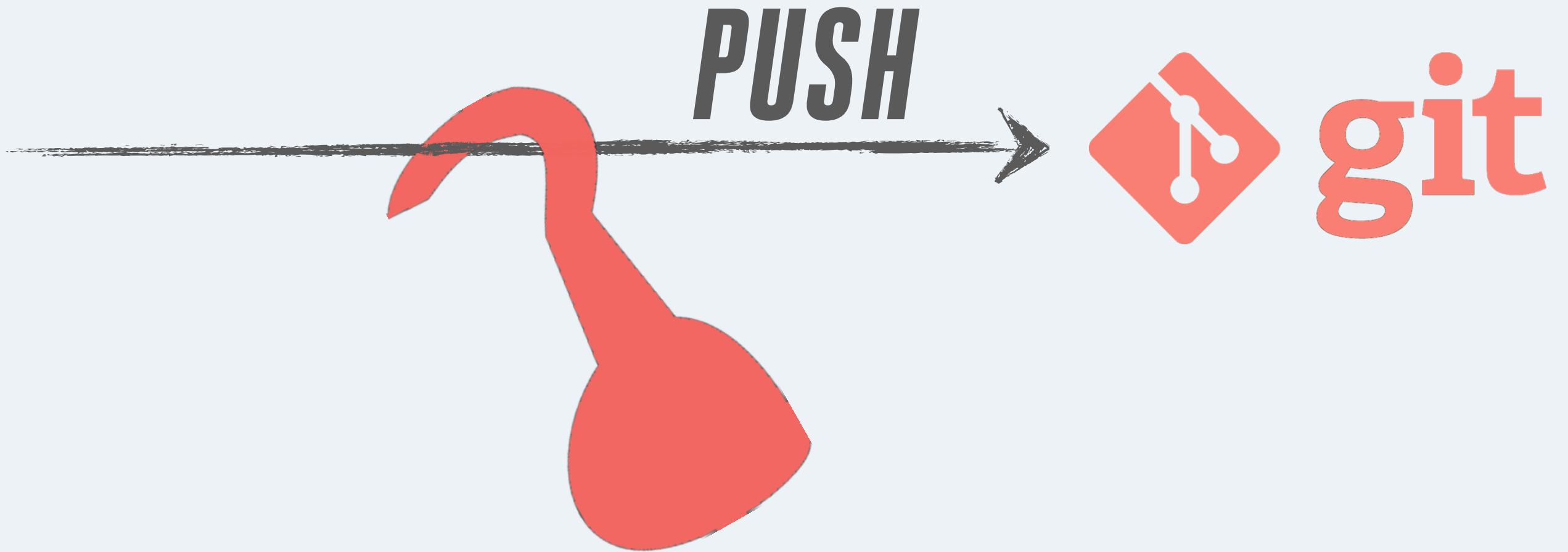


PUSH



git

Script is doing some stuff





**WATCH OUT:
HOOK MAGIC**

at Altasign

1

DON'T BRANCH FROM A RED BUILD



1

DON'T BRANCH FROM A RED BUILD



2

BE INFORMED



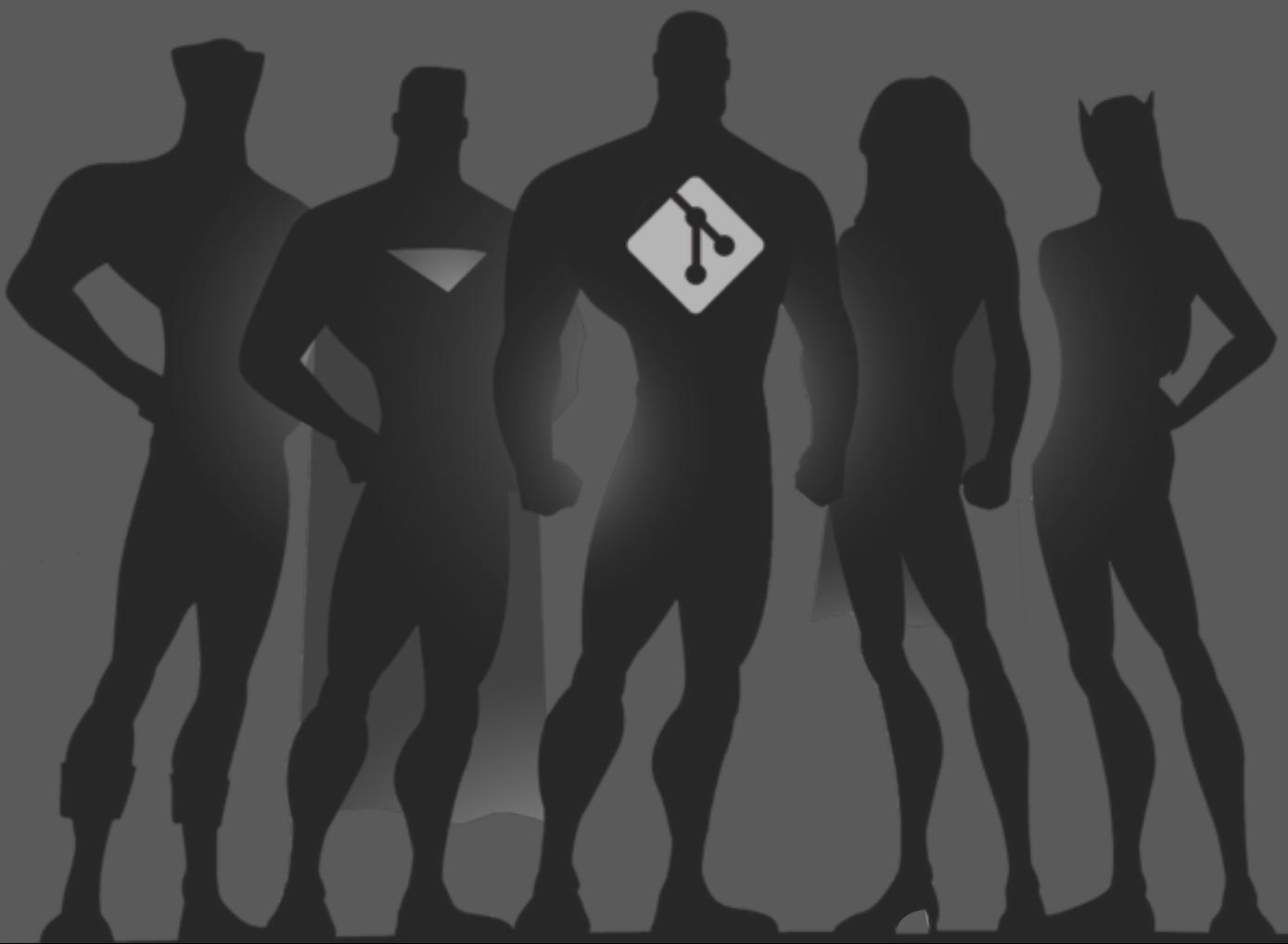
2

BE INFORMED

Bamboo	! Stash > Pull Request Build ›  STASHDEV-4781-changeset-icon-panel › #7 failed (rerun 2 times). 5 of 8818 failed. Manual run	10:33
Stash	Jonathan Poh committed to 1 branch at Stash/stash Created branch "STASHDEV-4954-add-missing-group-avatars"	10:38
Bamboo	✓ Stash > Pull Request Build ›  STASHDEV-4876-branch-model-default-branch › #3 passed. 7849 passed. Changes by Dariusz Kordonski	10:42
Stash	Dariusz Kordonski committed to 1 branch at Stash/stash On branch "STASHDEV-4876-branch-model-default-branch" - STASHDEV-4876 trying to fix the no-preload data case (b53fdfde432f)	10:44
Bamboo	! Stash > Master › #1563 failed. 4 of 5019 failed. Changes by Xu-Heng Tjhin	10:47
Stash	Lukasz Wlodarczyk has been a naughty dev and pushed some changes directly to master in JIRA/jira . These commits were unknown and should have been pushed on an issue branch first: - None marking the rename user metric unchecked. (ea555d6047f6)	Feb.-7 11:15 10:48
Bamboo	! Stash > Pull Request Build ›  STASHDEV-4954-add-missing-group-avatars › #1 failed. 7859 passed. First build for this plan	10:53
Stash	Dariusz Kordonski committed to 1 branch at Stash/stash On branch "master" - STASHDEV-4463 fixing branch model REST tests to accommodate for the revised e... (143c1fa6ba66)	10:57

WORKFLOWS WITH CI SERVERS

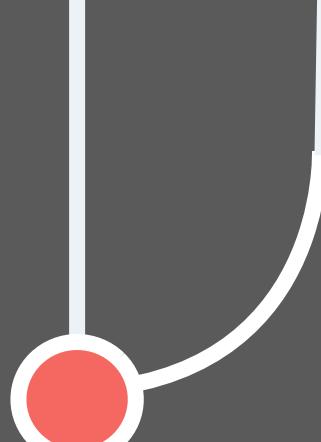
Continuous Integration ↴



WE WANT

A STABLE DEVELOP BRANCH THAT IS CI TESTED

DEVELOP



BUILDS ON FEATURE BRANCHES

this for sure

WE WANT

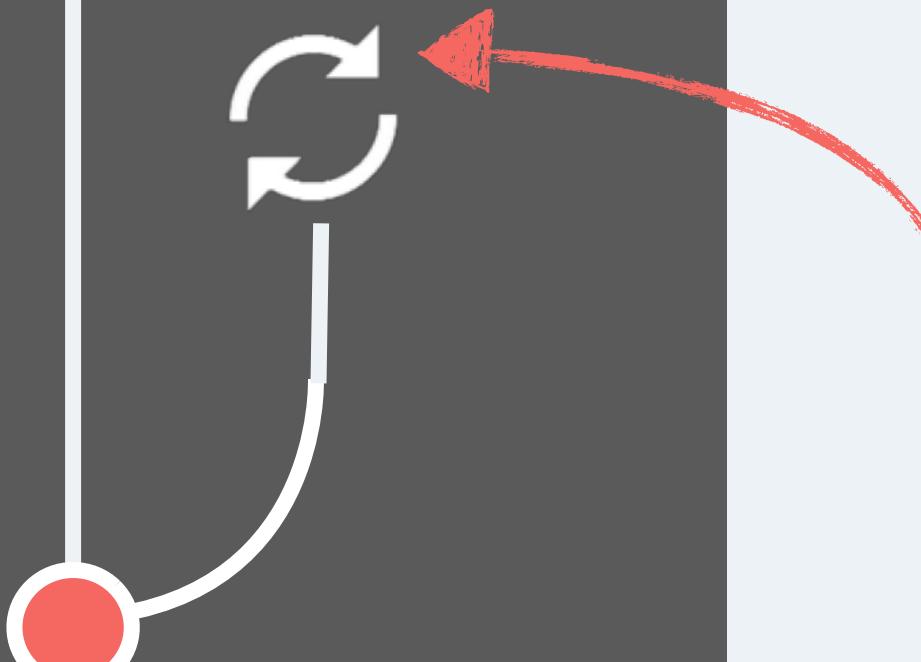
A STABLE DEVELOP BRANCH THAT IS CI TESTED

WE WANT

A STABLE FEATURE BRANCHES THAT ARE CI TESTED

to be sure, that we won't break things

DEVELOP



so we need this, too!

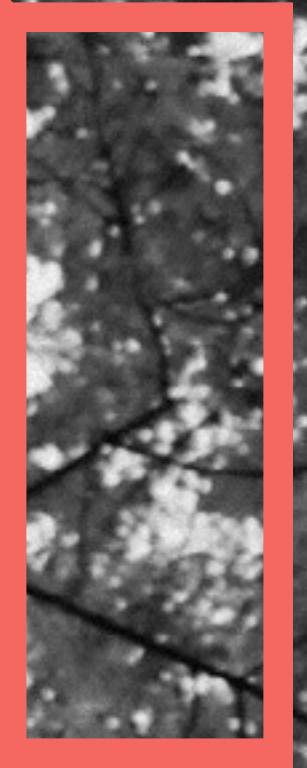
BUILDS ON FEATURE BRANCHES

YOU'LL FORGET TO SET IT UP!





AUTO DETECT BRANCHES



AUTO DETECT BRANCHES



BUT  + 

IS NOT ALWAYS



WE COULD

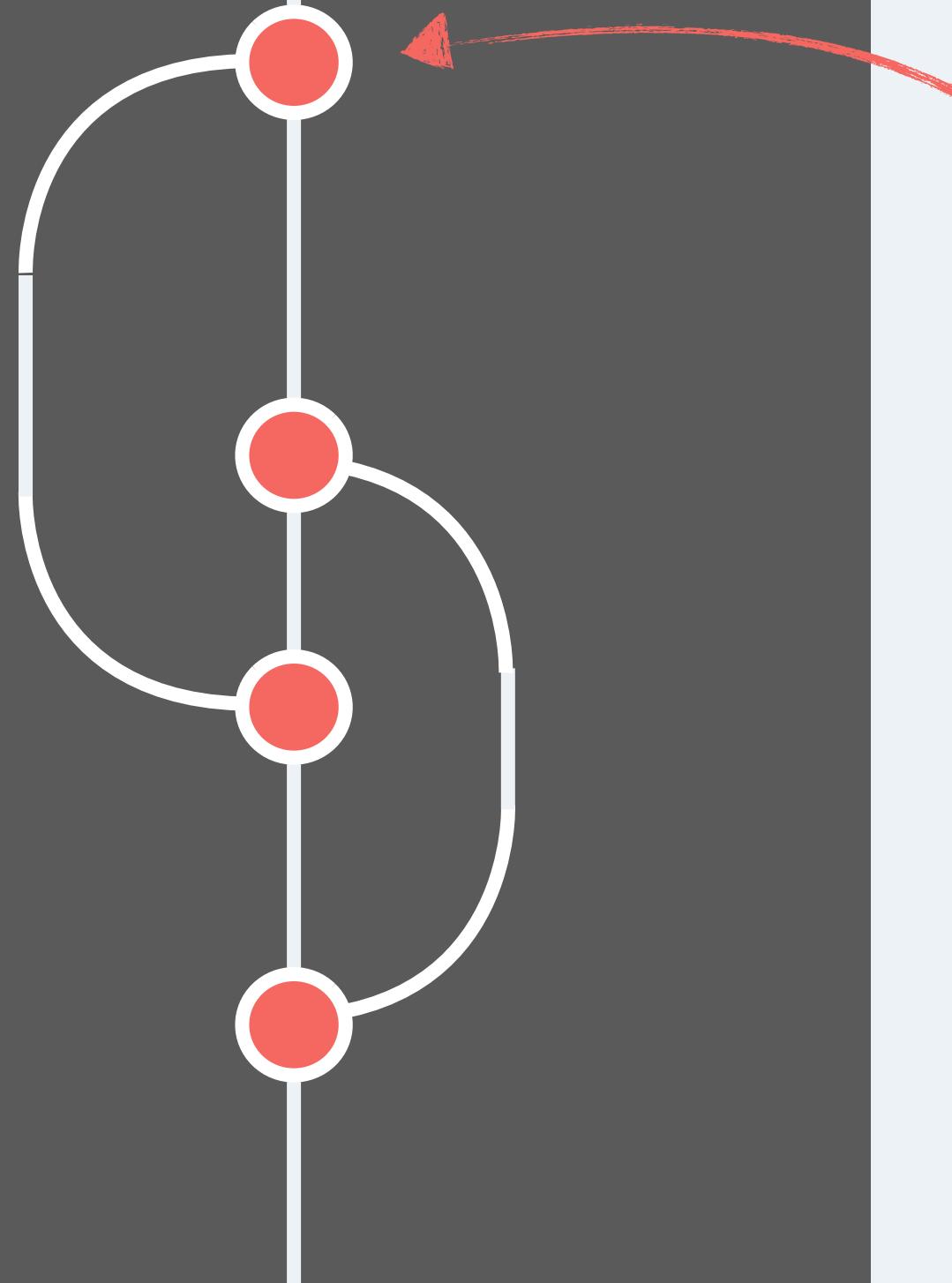
PULL CHANGES
BUILD LOCALLY
MERGE IF OK

WE COULD

MERGE & PRAY

WHY NOT USE
AUTO MERGE
ON
FEATURE BRANCHES ?

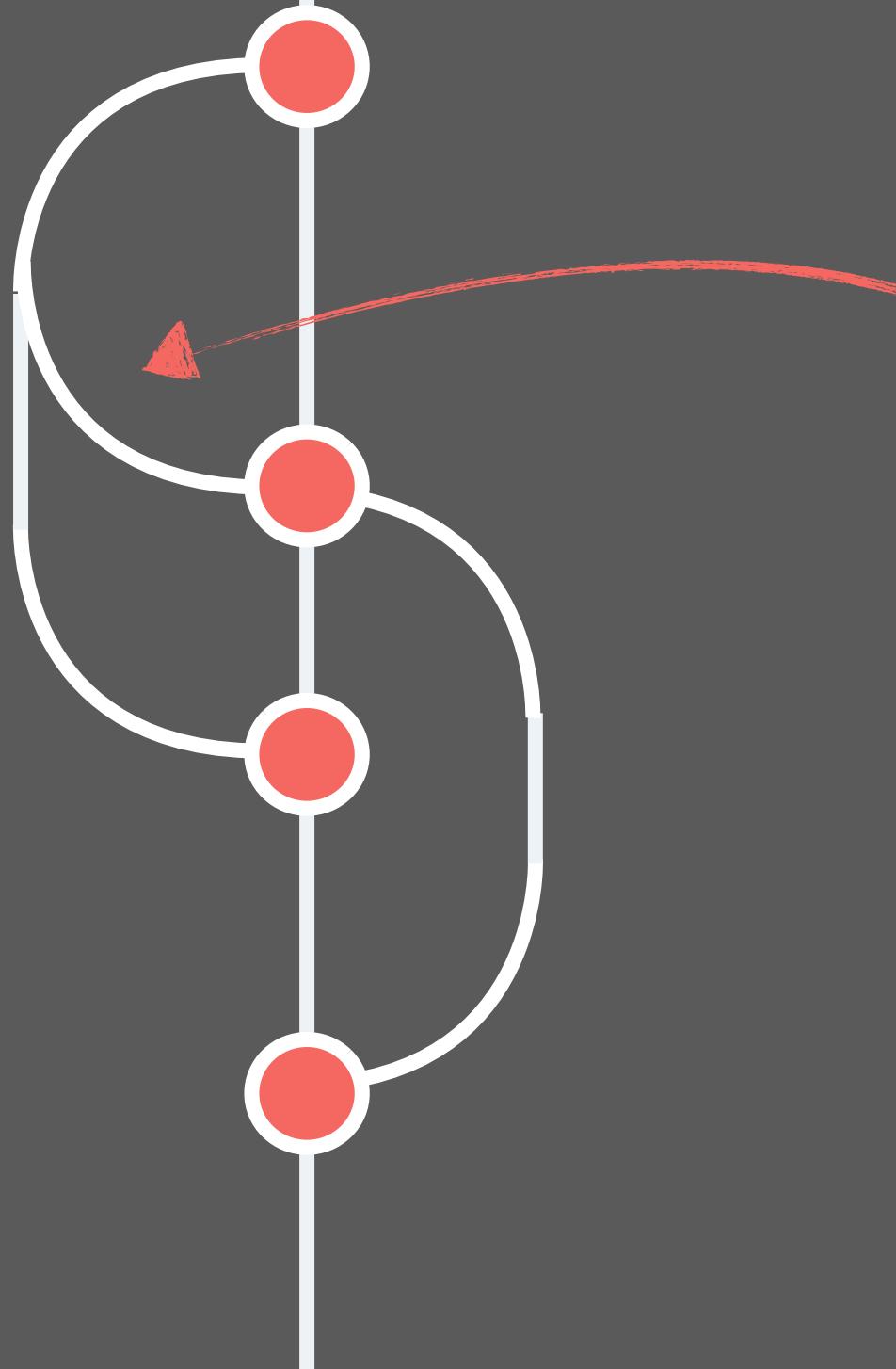
DEVELOP



AUTO MERGES

potential integration problem

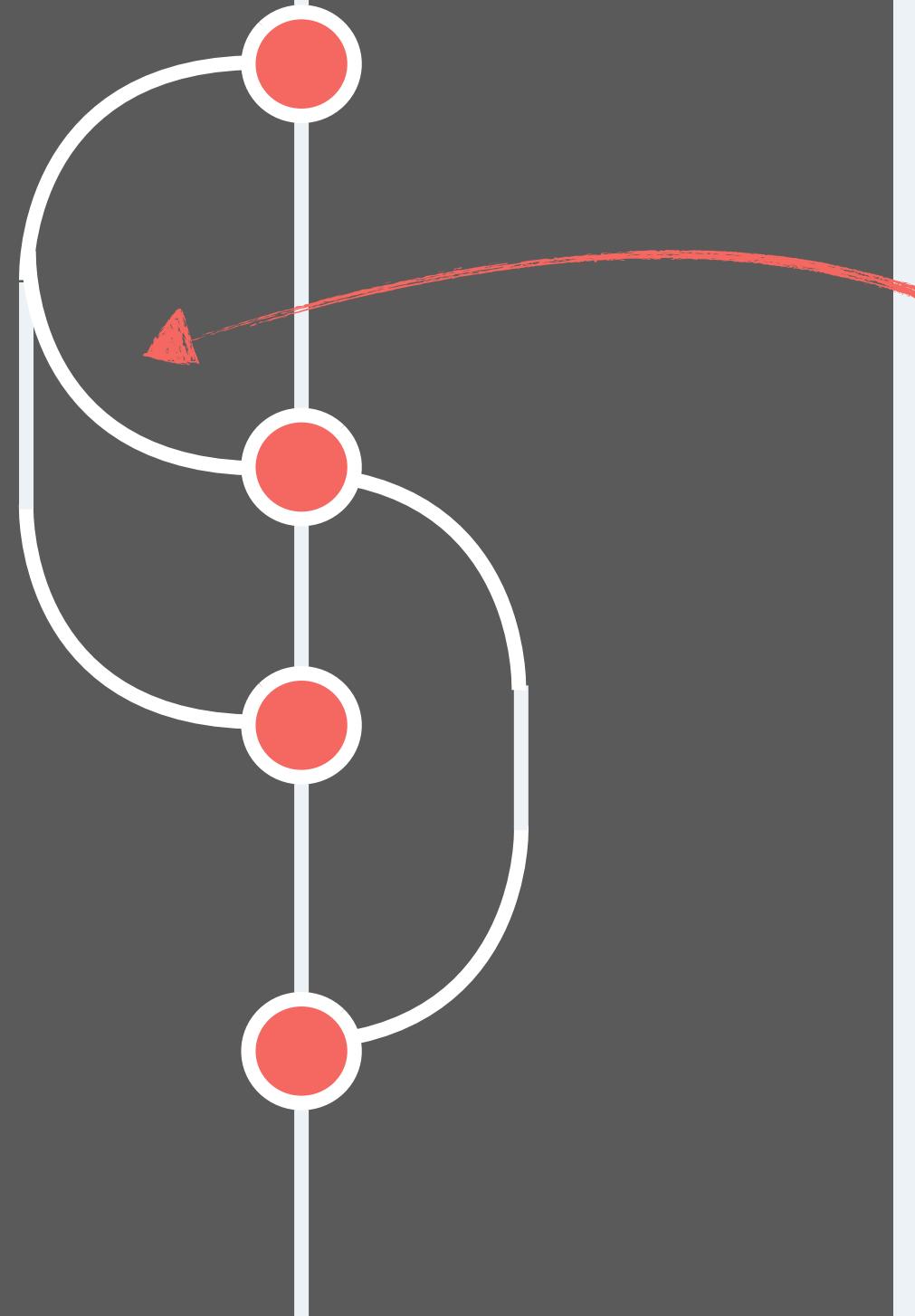
DEVELOP



AUTO MERGES

I told you: pull frequently

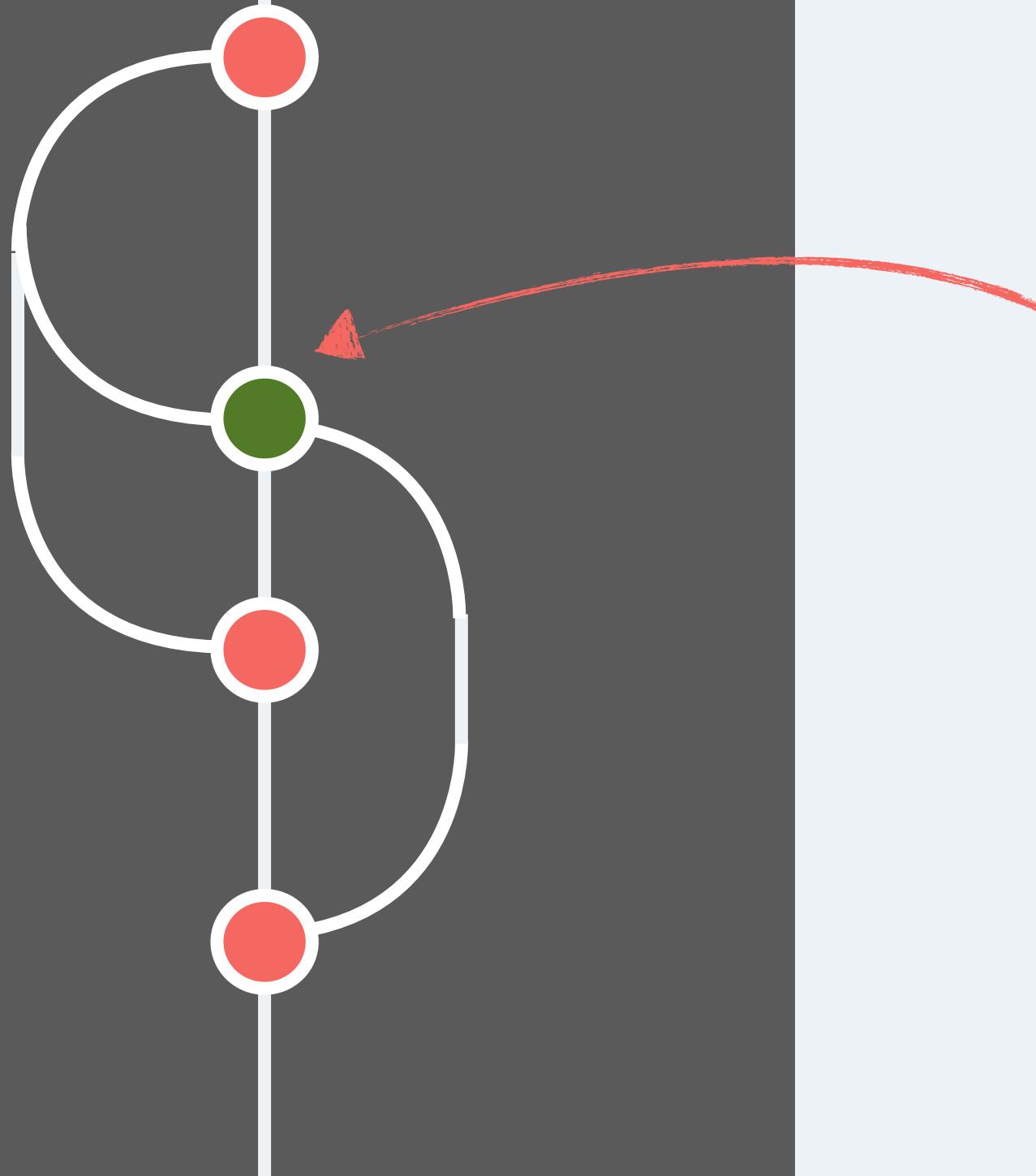
DEVELOP



AUTO MERGES

~~pull frequently automatically on commits~~

DEVELOP

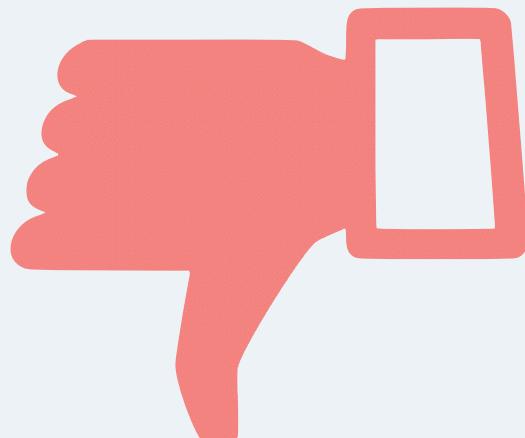


GREEN BUILD MERGES

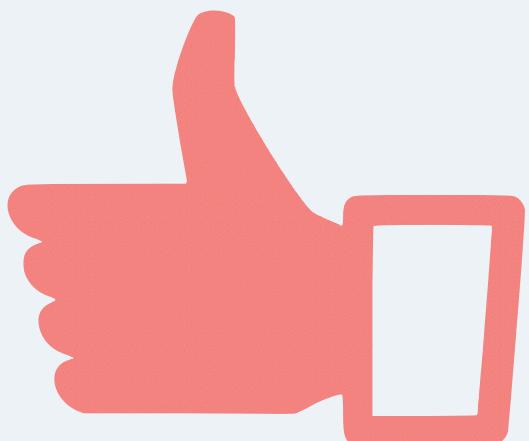
push automatically on green
builds of DEVELOP

BUILD ON EVERY PUSH KILLED OUR CI SERVERS

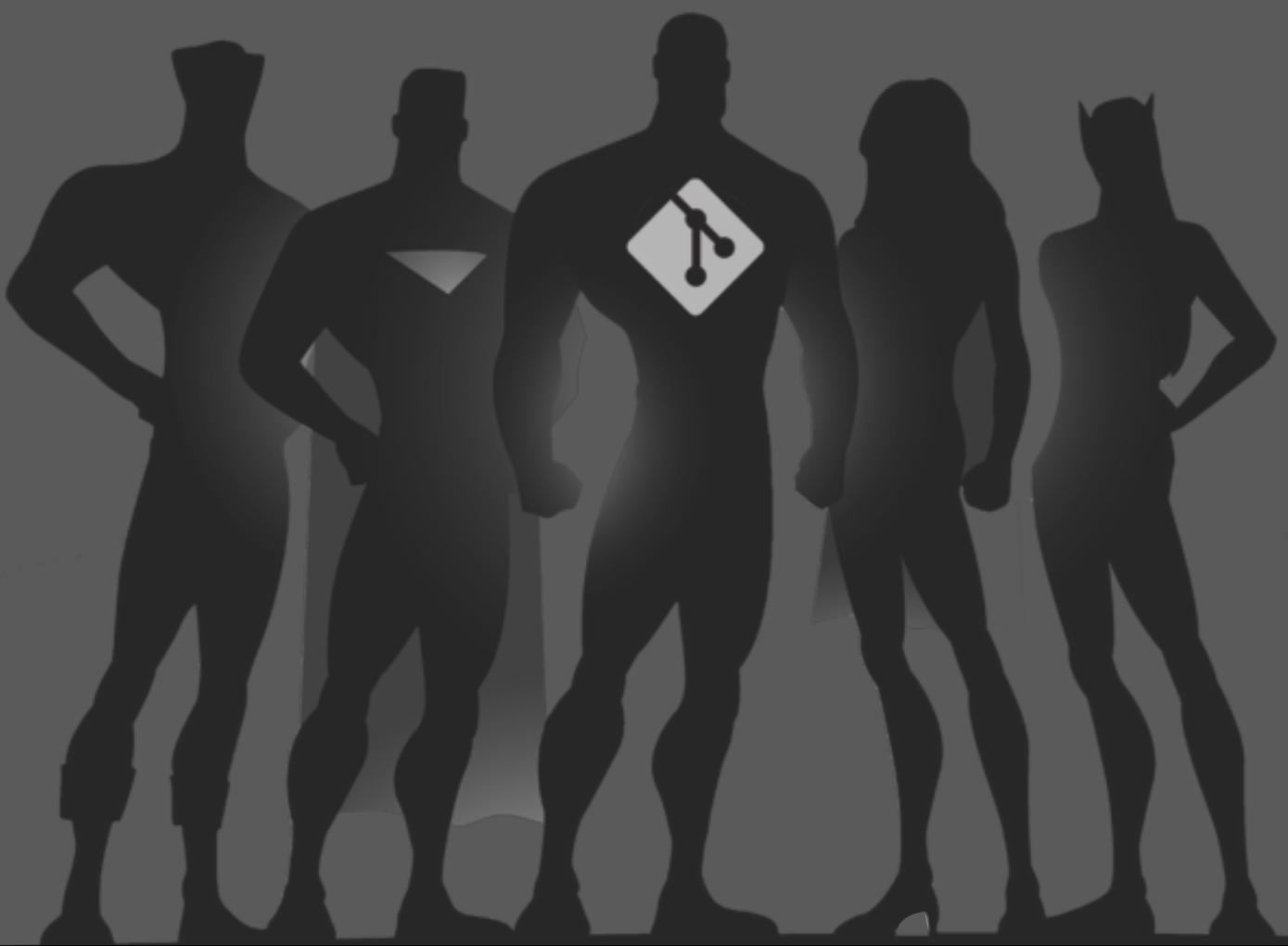
x features branches * x pushes a day
= server overload



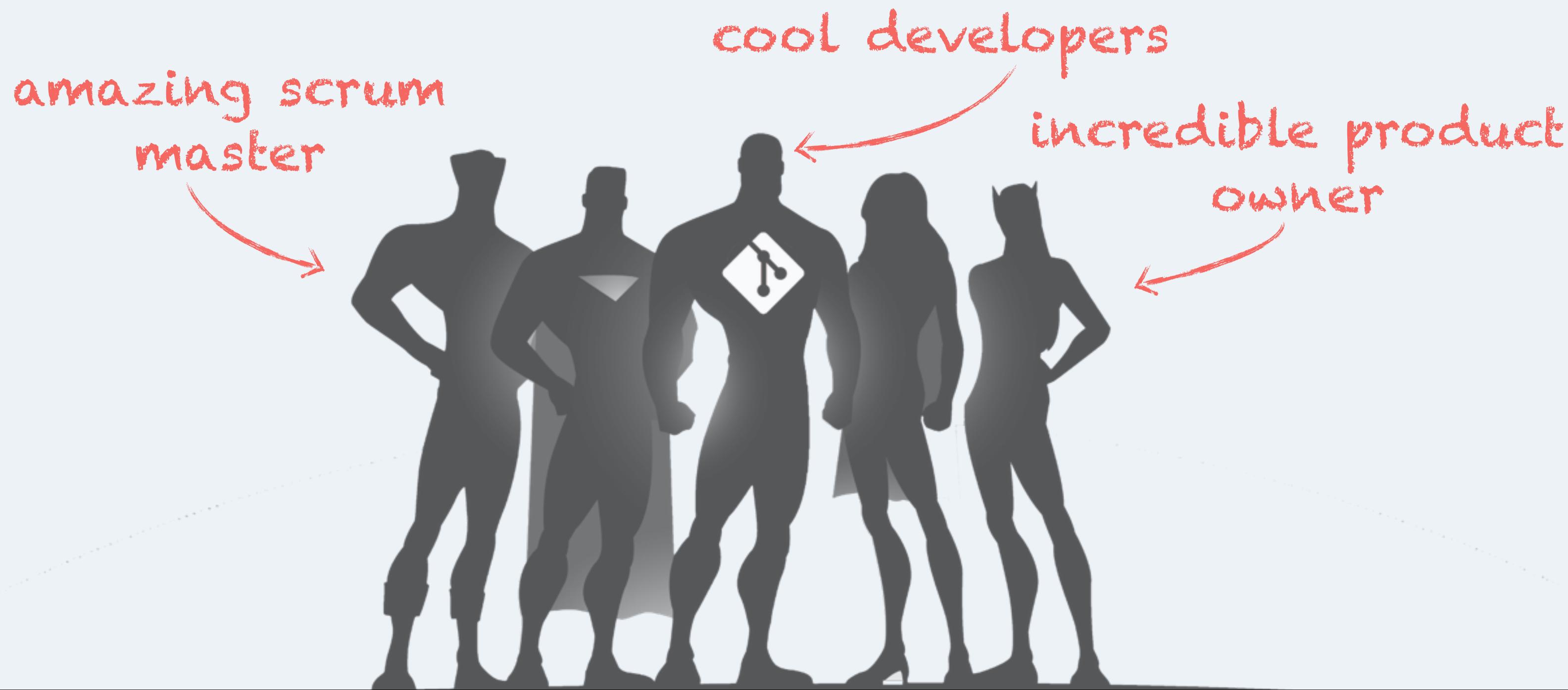
BUILD ON EVERY MERGE ON MASTER, RELEASE & DEVELOP
BUILD MANUALLY ON FEATURE BRANCHES
DOING AUTO MERGES ON GREEN BUILDS OF DEVELOP



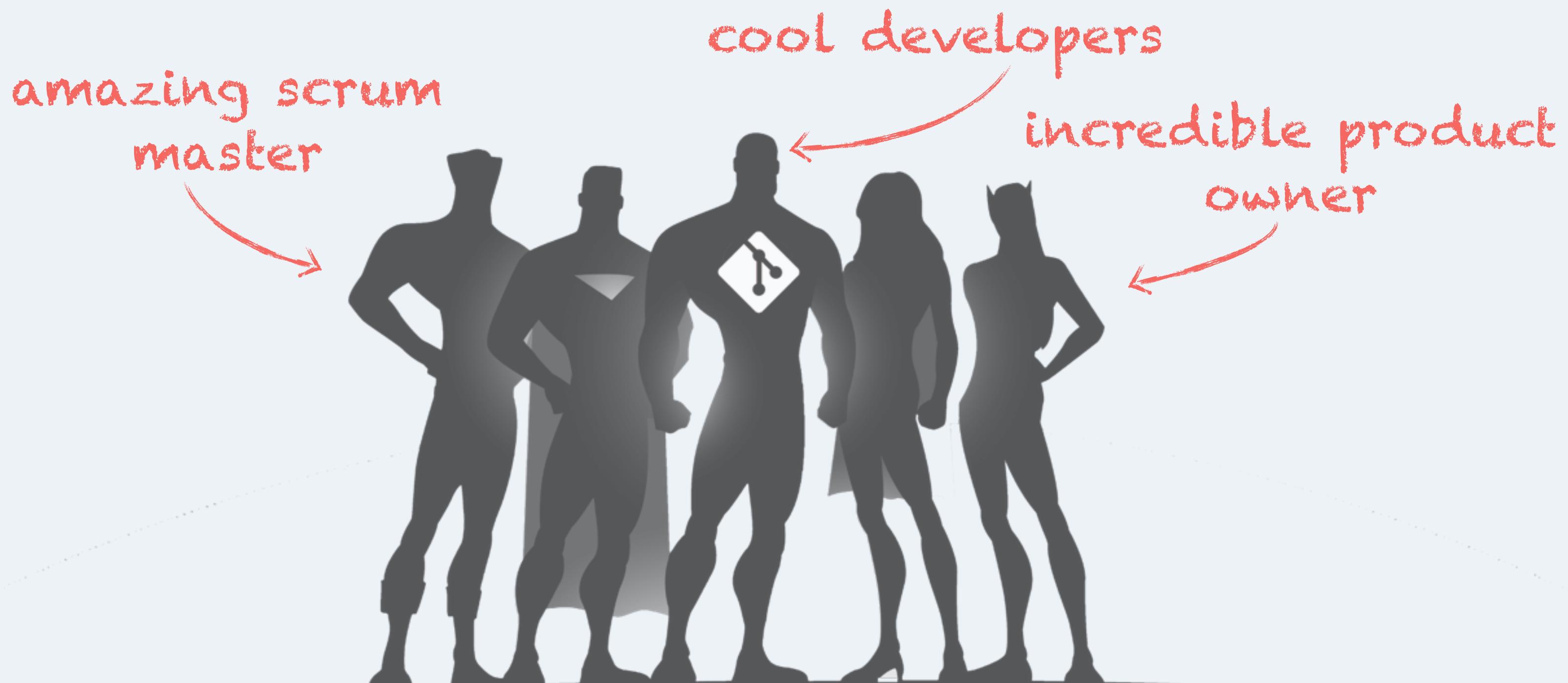
WORKFLOWS WITH ISSUE TRACKING



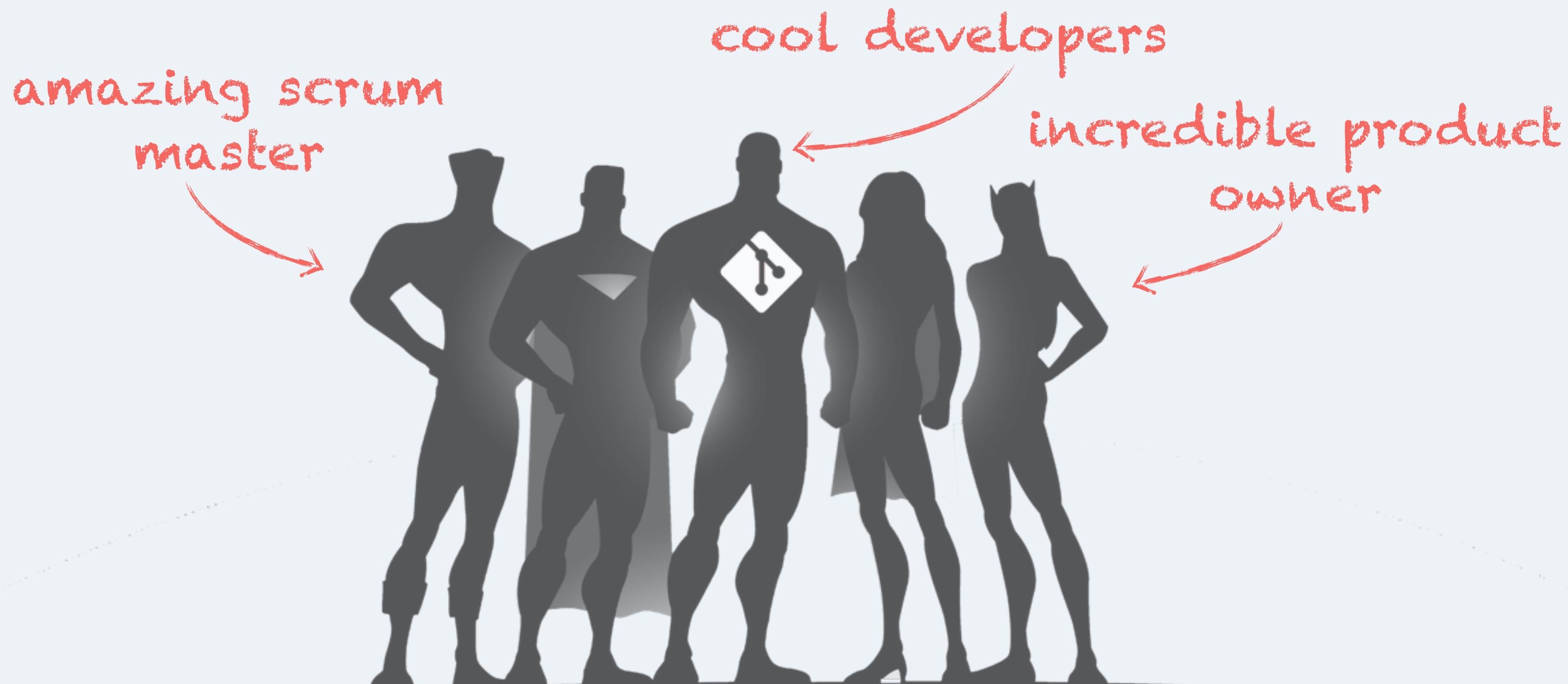
DIFFERENT **PEOPLE** HAVE DIFFERENT VIEWS ON YOUR **PROJECT**



ISSUE TRACKERS CONNECT US CODERS WITH THE **REAL WORLD**



ISSUE TRACKERS CONNECTS GIT WITH THE REAL WORLD



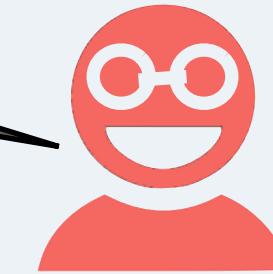
WE LIVE IN CODE!

LET ME WORK IN CODE
I JUST NEED TO KNOW WHAT
MY CODE SHOULD SOLVE!



HOW DO ISSUE TRACKERS CONNECT TO THE CODE?

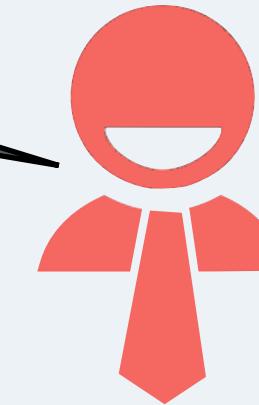
Problem understood,
I start coding



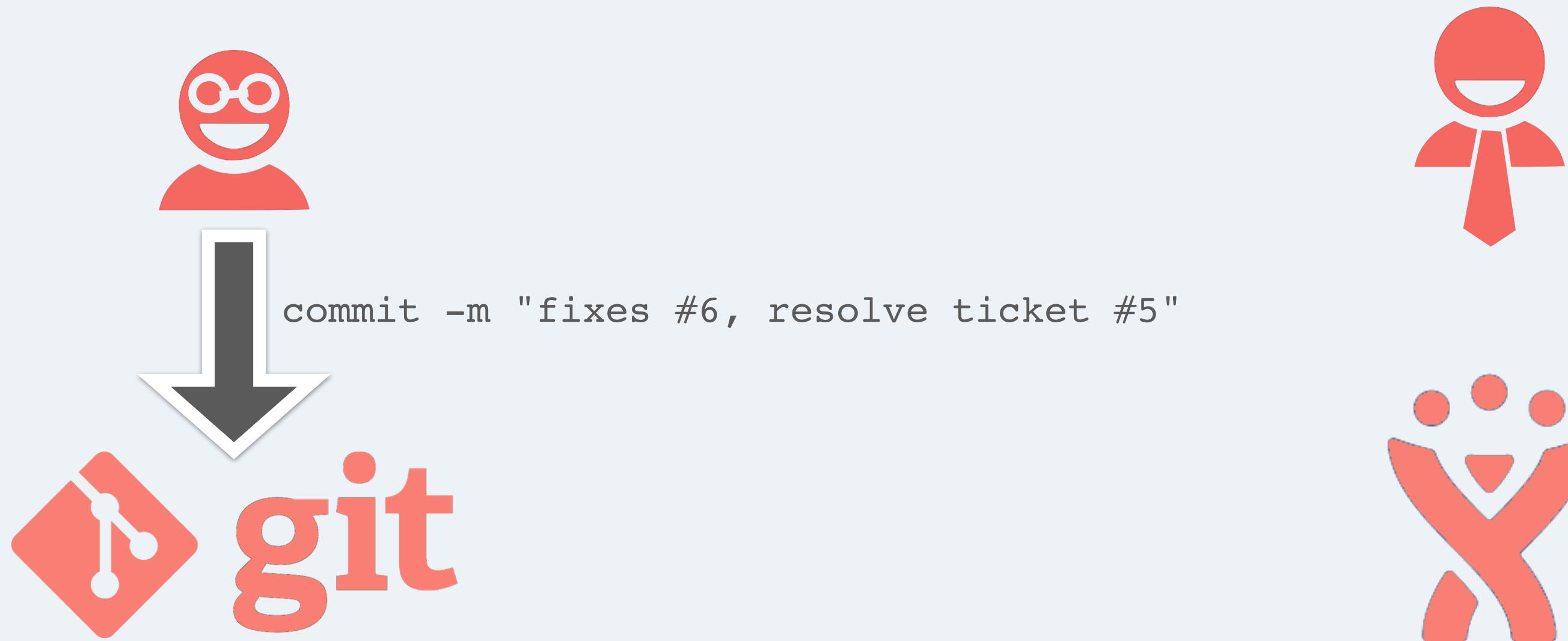
HOW DO ISSUE TRACKERS CONNECT TO THE CODE?



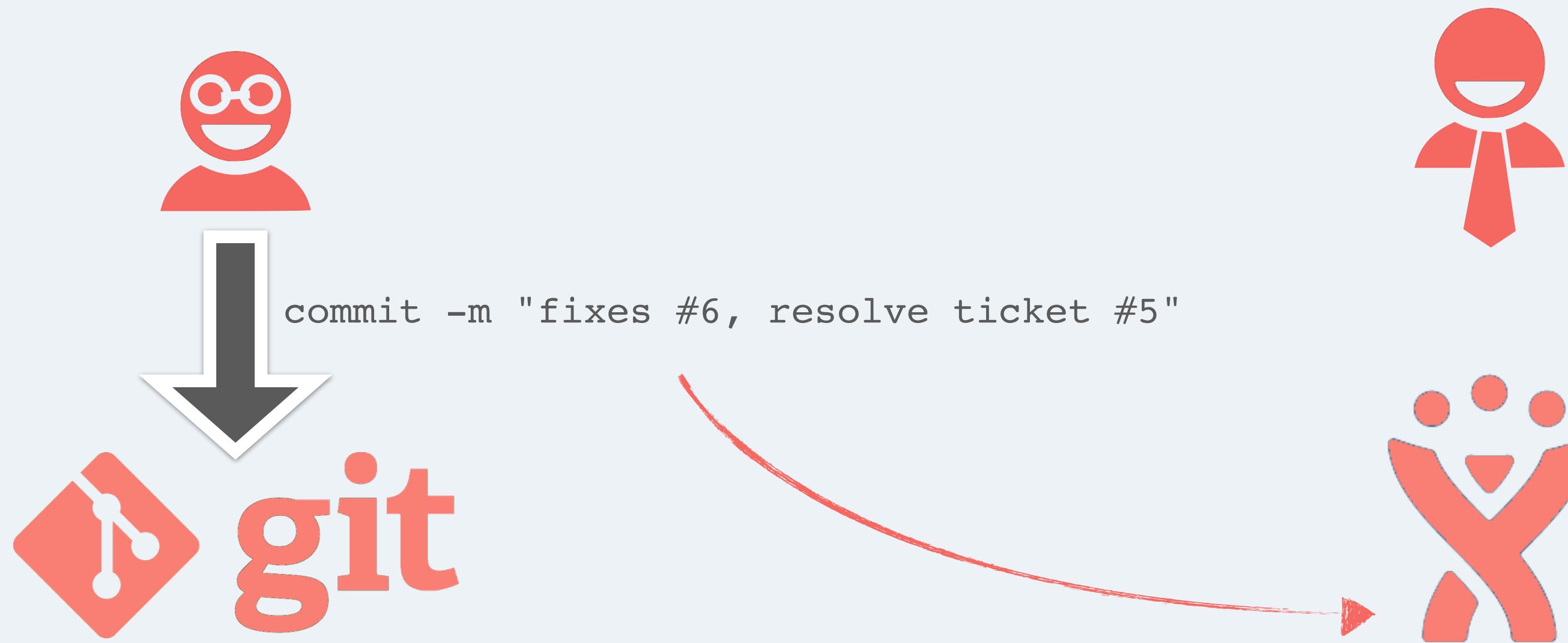
Ah good,
the coders started



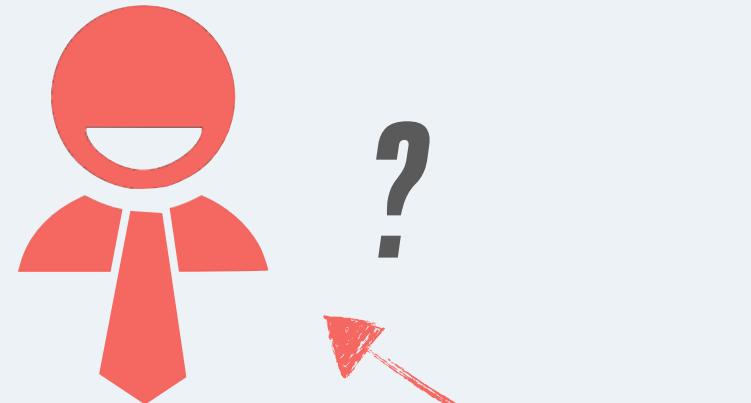
HOW DO ISSUE TRACKERS CONNECT TO THE CODE?



HOW DO ISSUE TRACKERS CONNECT TO THE CODE?



WHAT'S IN IT FOR



non geeks

SEE WHAT'S IN *DEVELOP*

SEE ISSUES IN *REVIEW*

SEE *FINISHED* ISSUES ALREADY MERGED WITH *DEVELOP*

WHAT'S IN IT FOR



?

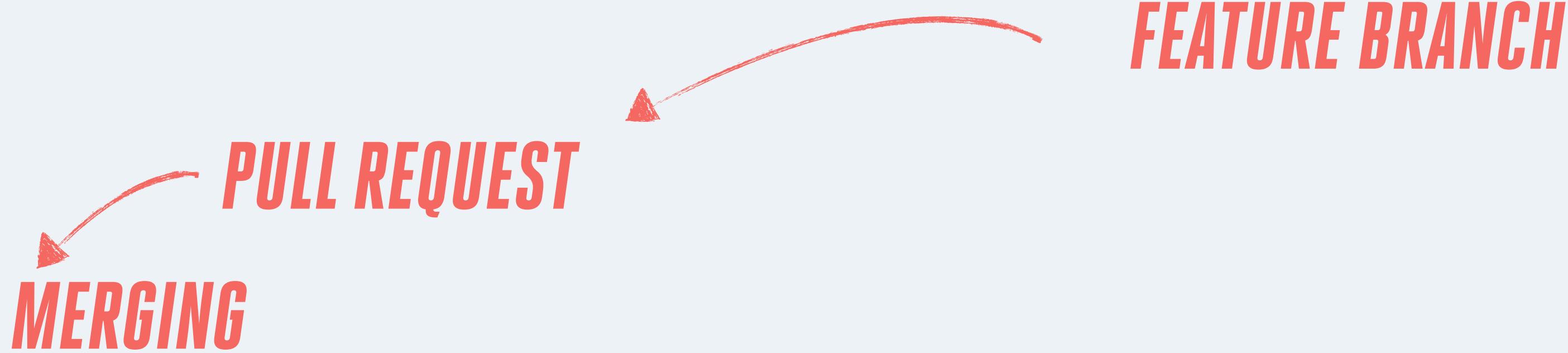
DOING CODE REVIEWS IN CONTEXT

BROWSING FEATURE DESCRIPTIONS SHOW CODE CHANGES

... I HAVE A DREAM



GIT WORKFLOW



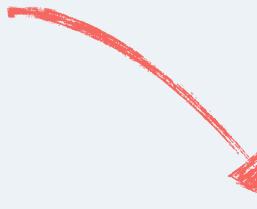
ISSUE WORKFLOW

'START PROGRESS'



'IN REVIEW'

'RESOLVED'



'START PROGRESS'

PULL REQUEST

MERGING

FEATURE BRANCH

'IN REVIEW'

'RESOLVED'

GIT COMBINED ISSUE WORKFLOW

'START PROGRESS' ON AN ISSUE CREATES A *FEATURE BRANCH*

DOING A *PULL REQUEST* MARKS THE ISSUE '*IN REVIEW*'

***MERGING* A *FEATURE BRANCH* MARKS THE ISSUE AS '*RESOLVED*'**



RISE OF THE GIT TOOLS

**DON'T FLOW
LIKE IT'S 1999**



THANKS

svenpet.com/talks
 *@svenpet*

RESOURCES

GIT WORKFLOWS

<https://www.atlassian.com/git/workflows>

AUTO MERGES

<http://blogs.atlassian.com/2013/05/git-automatic-merges-with-server-side-hooks-for-the-win/>

GIT RESOURCES

<https://www.atlassian.com/git/resources>

FORKS & UPSTREAMS

<https://blogs.atlassian.com/2013/07/git-upstreams-forks/>

PICTURES

<http://www.flickr.com/photos/43322231@N07/4390305784/>
<http://www.flickr.com/photos/40011478@N00/2890078135/>
<http://www.flickr.com/photos/33909206@N04/7638306424>
<http://www.flickr.com/photos/8340753@N04/501465437/>
<http://www.flickr.com/photos/33774513@N08/3510825685/>
<http://www.flickr.com/photos/55723329@N00/3631928376/>
<http://www.flickr.com/photos/11389536@N03/5235804240/>
<http://www.flickr.com/photos/43322231@N07/5410534167>
<http://www.flickr.com/photos/43322231@N07/5366609719>
<http://www.flickr.com/photos/43322231@N07/4205372249/>
<http://www.flickr.com/photos/43322231@N07/4466599835/>
<http://www.flickr.com/photos/45940879@N04/5598027415/>
<http://www.flickr.com/photos/83346641@N00/3517813158/>
<http://www.flickr.com/photos/18382722@N00/4163812132/>

<http://www.flickr.com/photos/80384851@N00/2653886484/>
<http://www.flickr.com/photos/21461615@N00/1353110751>
<http://www.flickr.com/photos/82402697@N00/449512594>
<http://www.flickr.com/photos/7228825@N05/675810372>
<http://www.flickr.com/photos/38451115@N04/4016842259/>
<http://www.flickr.com/photos/8749778@N06/5601758699/>
<http://www.flickr.com/photos/37831703@N00/3391877877>
<http://www.flickr.com/photos/60648084@N00/2900765643/>
<http://www.flickr.com/photos/28415633@N00/8401940159>
<http://www.flickr.com/photos/41304517@N00/6814372181>
<http://www.flickr.com/photos/99796131@N00/120468504/>
<http://www.flickr.com/photos/92269745@N00/4612732045/>
<http://www.flickr.com/photos/57768341@N00/3800568262/>
<http://www.flickr.com/photos/15923063@N00/3367228632/>