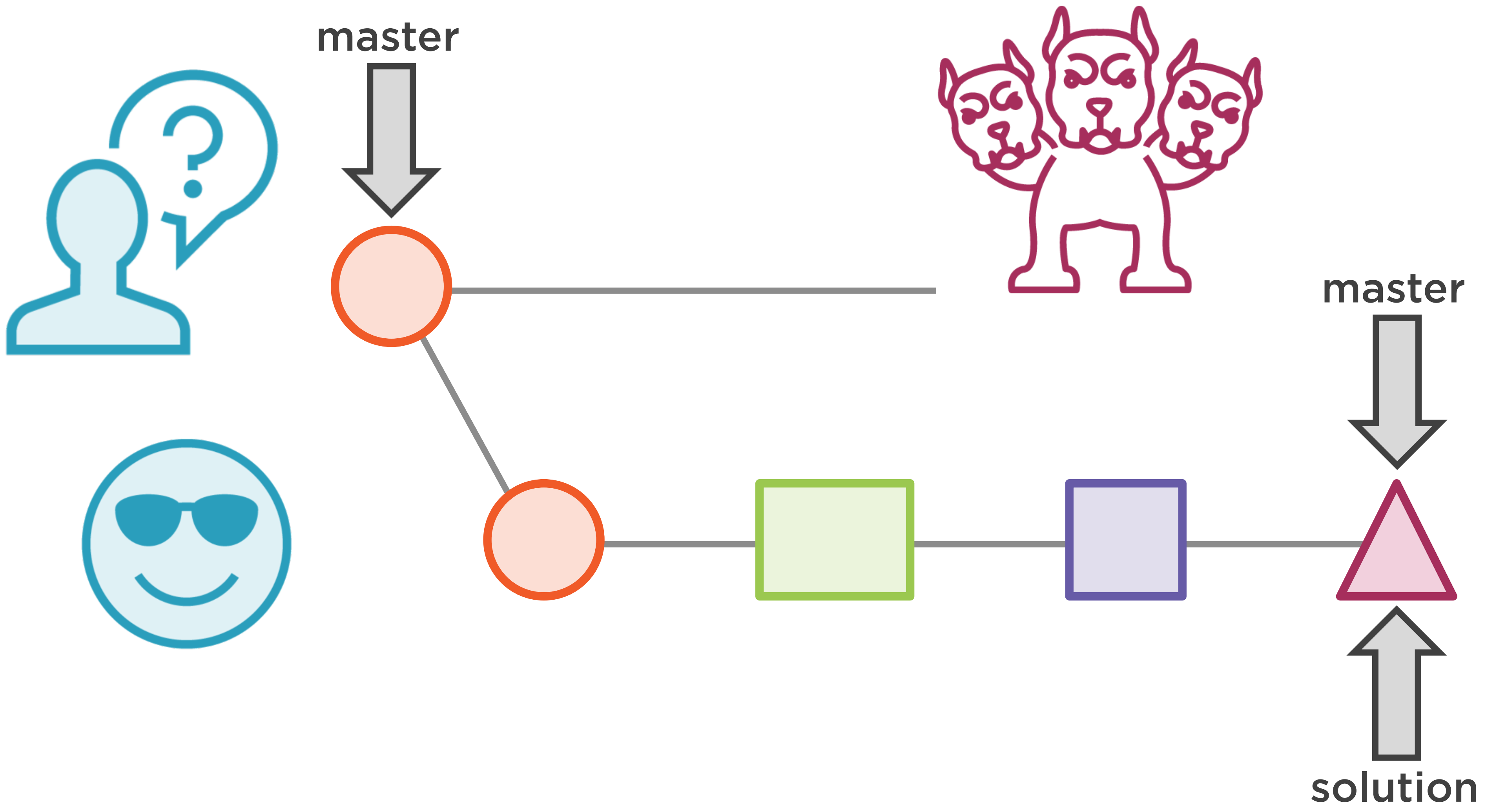
Comparing and Merging Branches

Craig Golightly

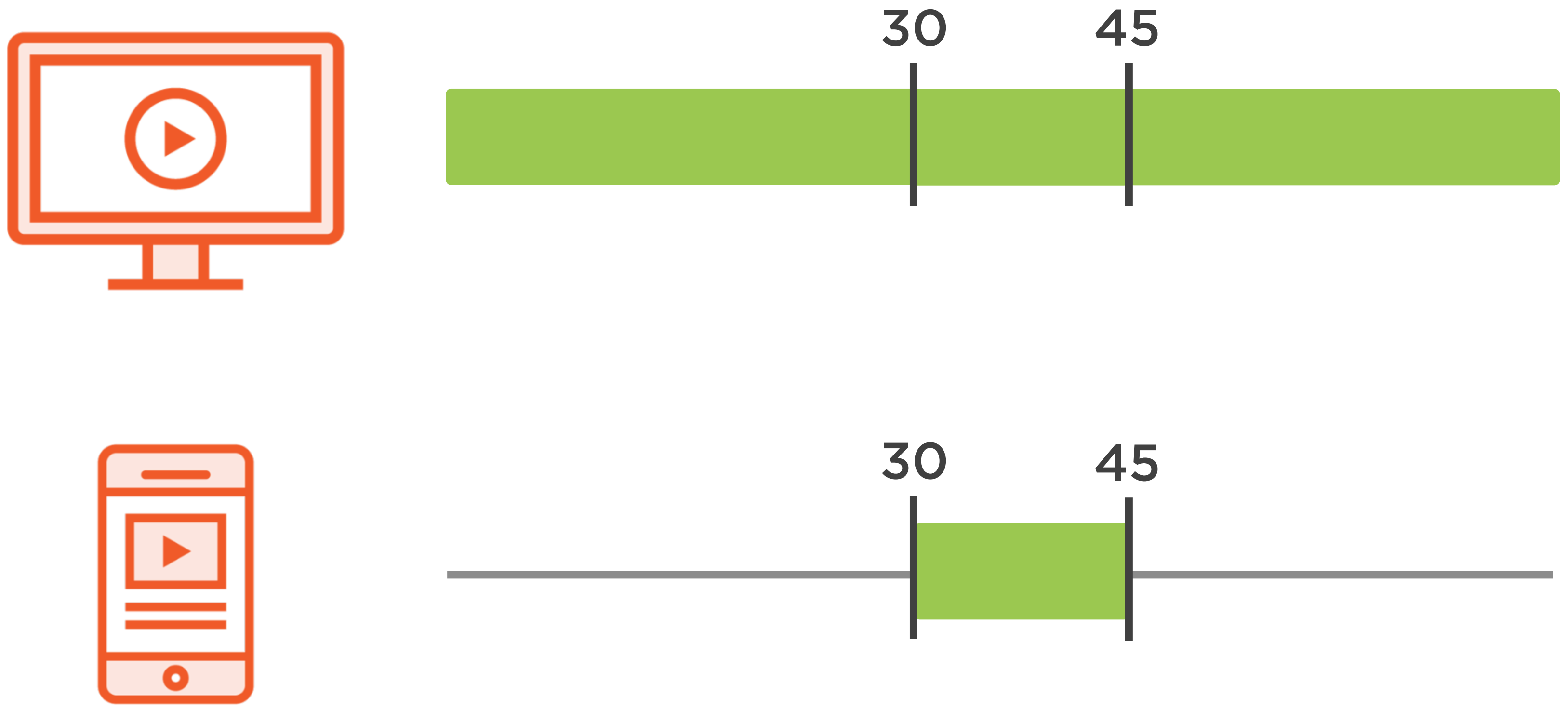
SENIOR SOFTWARE CONSULTANT

@seethatgo   www.seethatgo.com

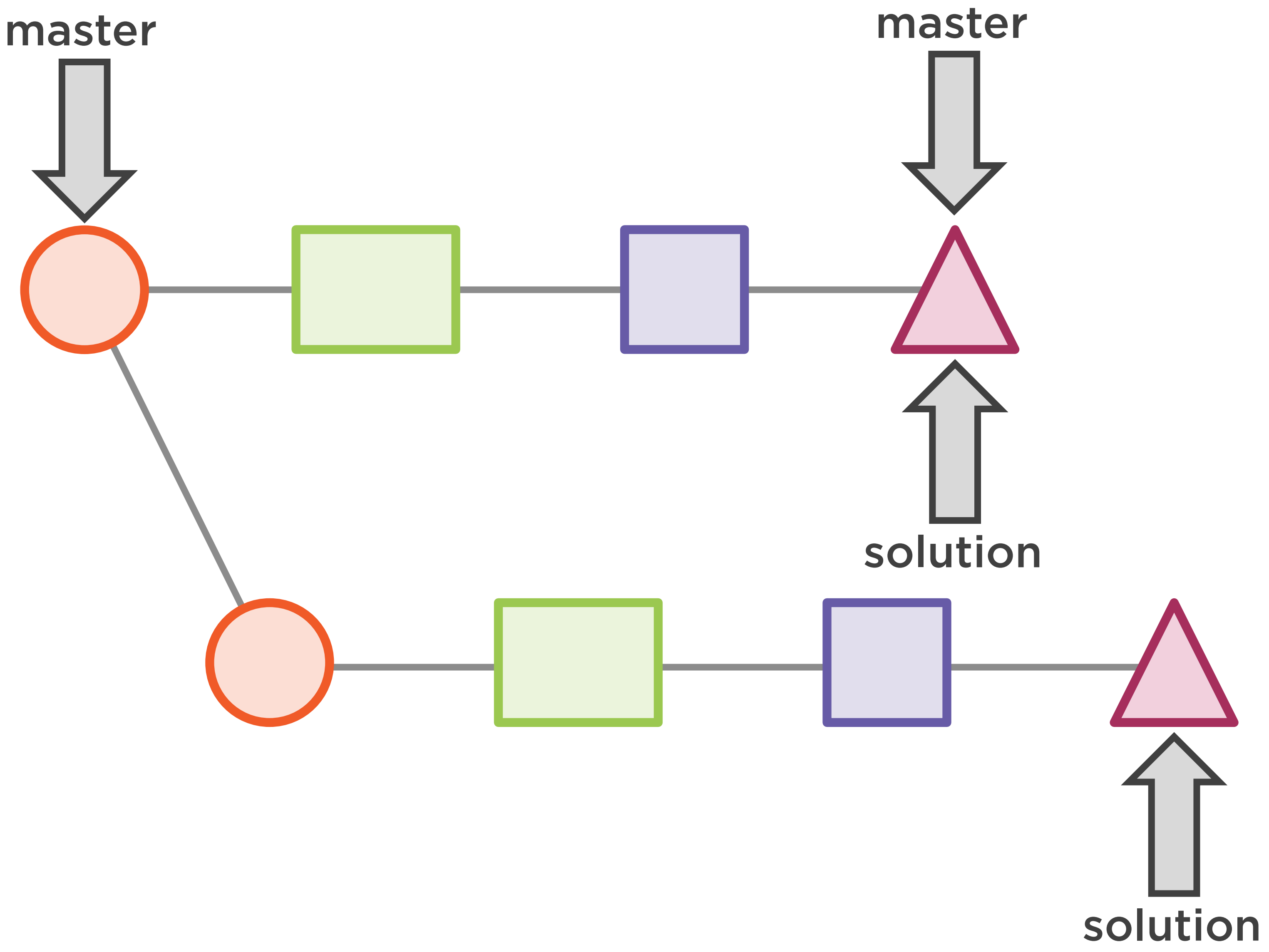
Fast Forward



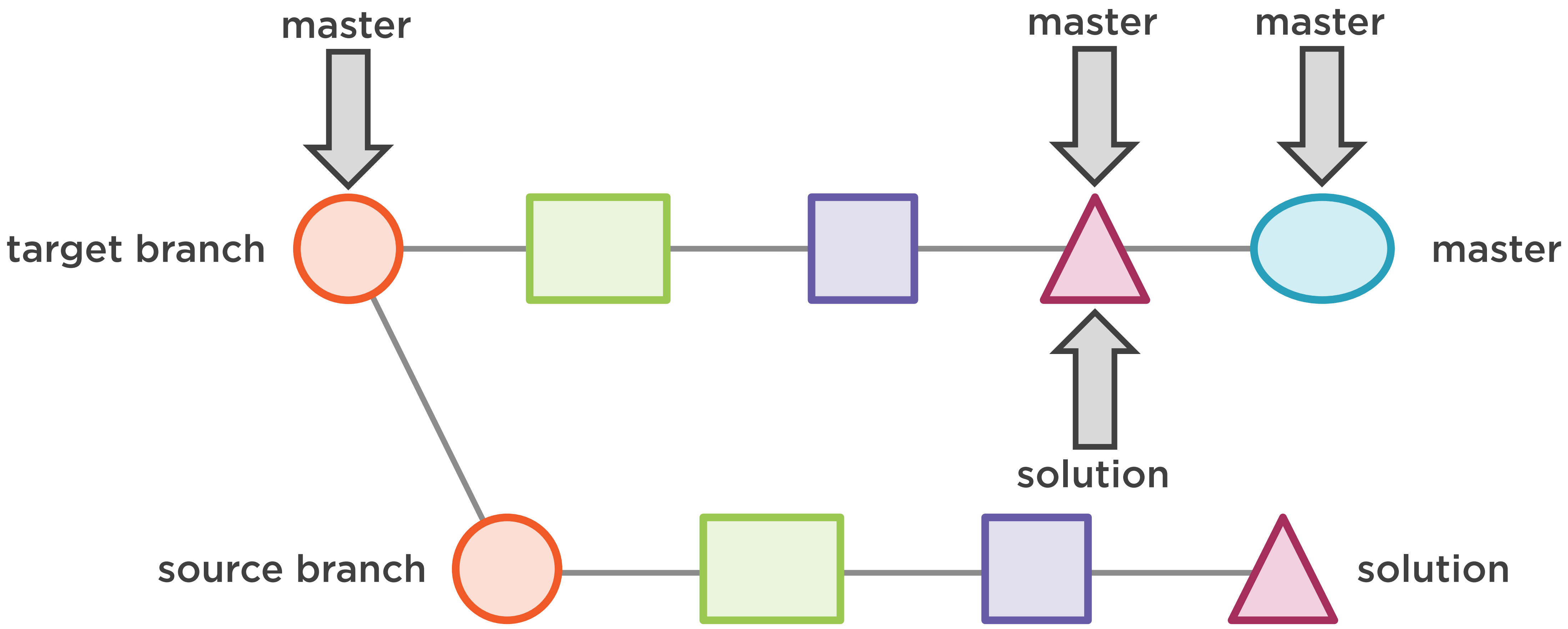
Movie on Multiple Devices



Fast Forward



Target and Source



```
git checkout <target-branch>

git merge <source-branch>

git checkout master

git merge ticket1
```

Merge Branches

Switch to target branch

Merge source branch

What’s Going to Happen?

Can you preview a merge before you do it?

```
git diff <branch1> <branch2>

git diff master ticket1
```

Compare Branches

Show changes between tips of branches

Does not change anything

More to come!

git rebase

Clean up local history
- Focus on end result

Should increase accuracy and clarity

Rough draft vs. final copy

Rebase is an advanced feature
- Not mandatory
- It can cause problems
- Times to NOT use it

Rebase

Do not rebase a public branch

Can cause confusion and lost work

Team guidelines

Check with your team about using rebase

Rebasing branches

“Rewriting Git History” course is a deep dive into rebase

Rebase Scenarios

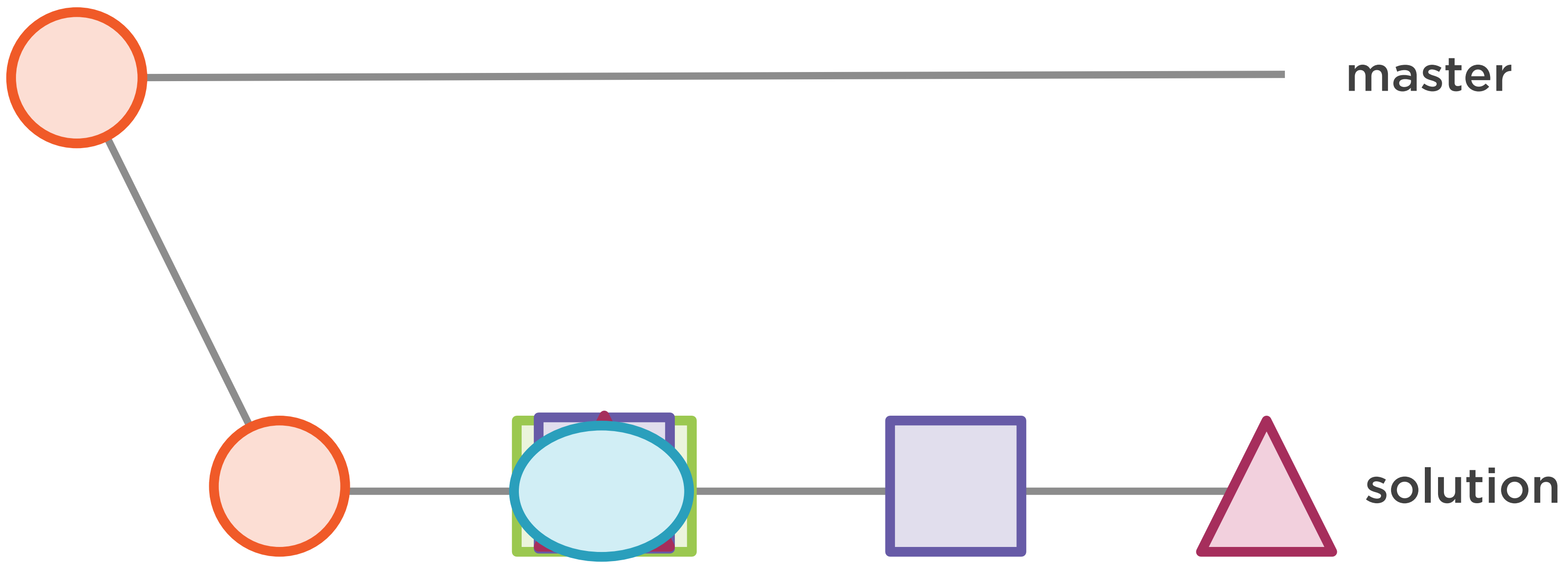
Clean up history

Clean up your local history before sharing a branch

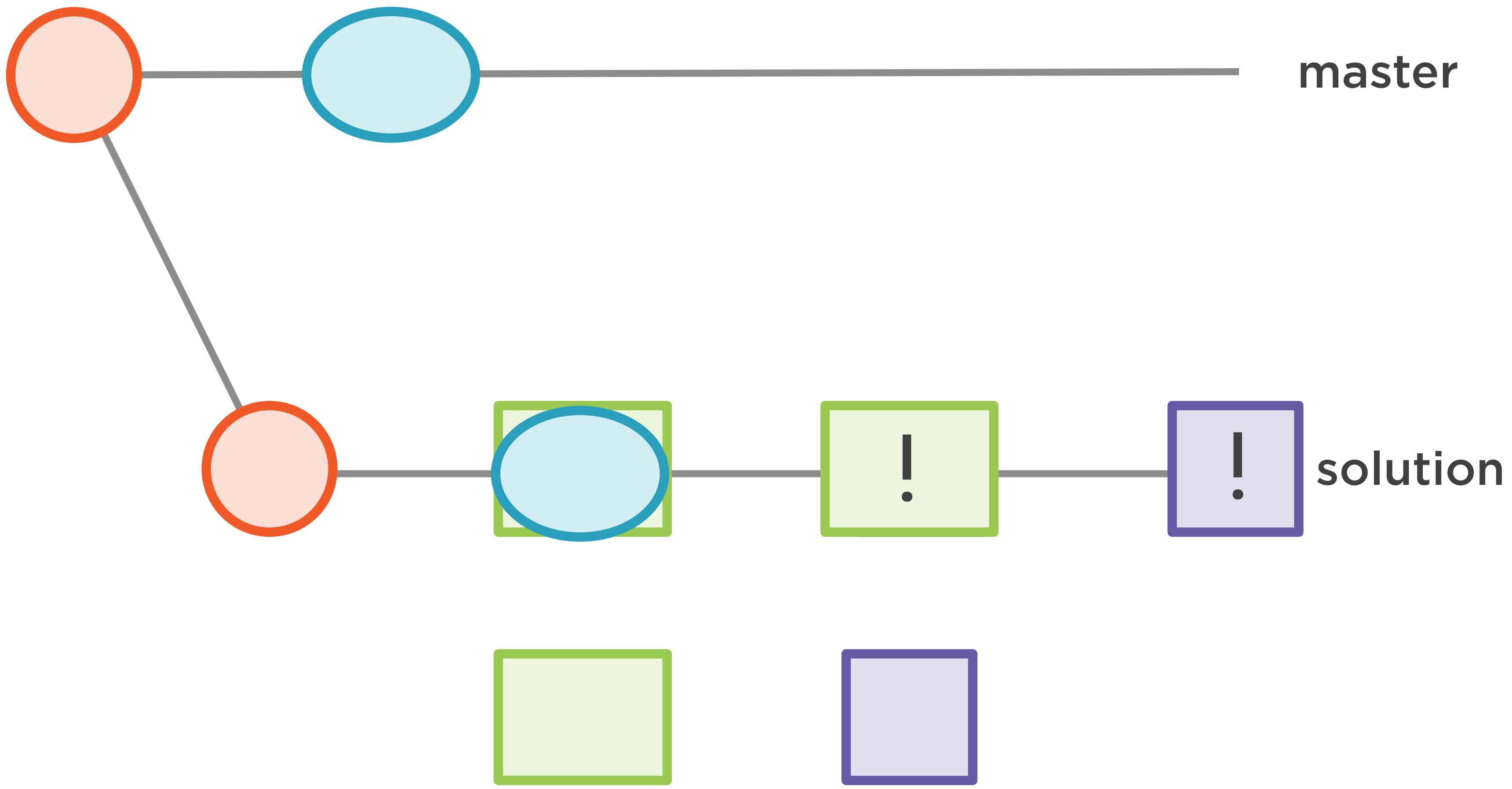
Pull without merge

Pull changes from a branch into your branch without performing a merge

Squash Commits



Rebase Branch from Master



```
git log --oneline
```
◀ See the branch history

```
git merge-base ticket1 master
```
◀ Get the original base of the “ticket1” branch created from master

```
git rebase -i <commit-sha>
```
◀ Start the rebase from the commit sha

```
pick 1285e6c starting f1
squash b714e68 more work in f1
squash 985d1e1 completed f1
```
◀ Squash the commits you want to combine into a single commit

```
# This is the 1st commit message:
starting f1
# This is the commit message #2:
doing more work in f1
# This is the commit message #3:
completed feature f1
```
◀ Choose the commit message you want to use

```
git rebase master
```
◀ Pull in changes from master then replay branch commits after

Copy specific commits to another branch

Bugfix for multiple versions of product

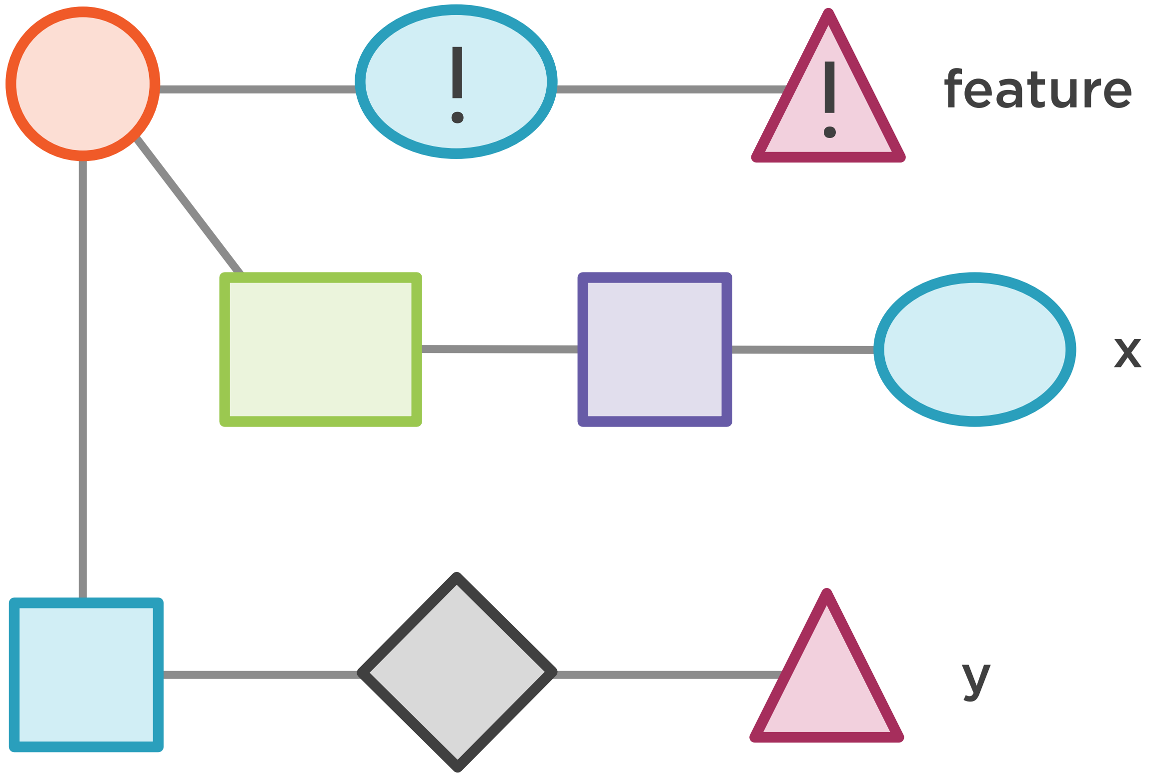
Capture commits from inactive branch

Move specific commits to your branch
- Features needed for your ticket
- Branches not ready to merge yet

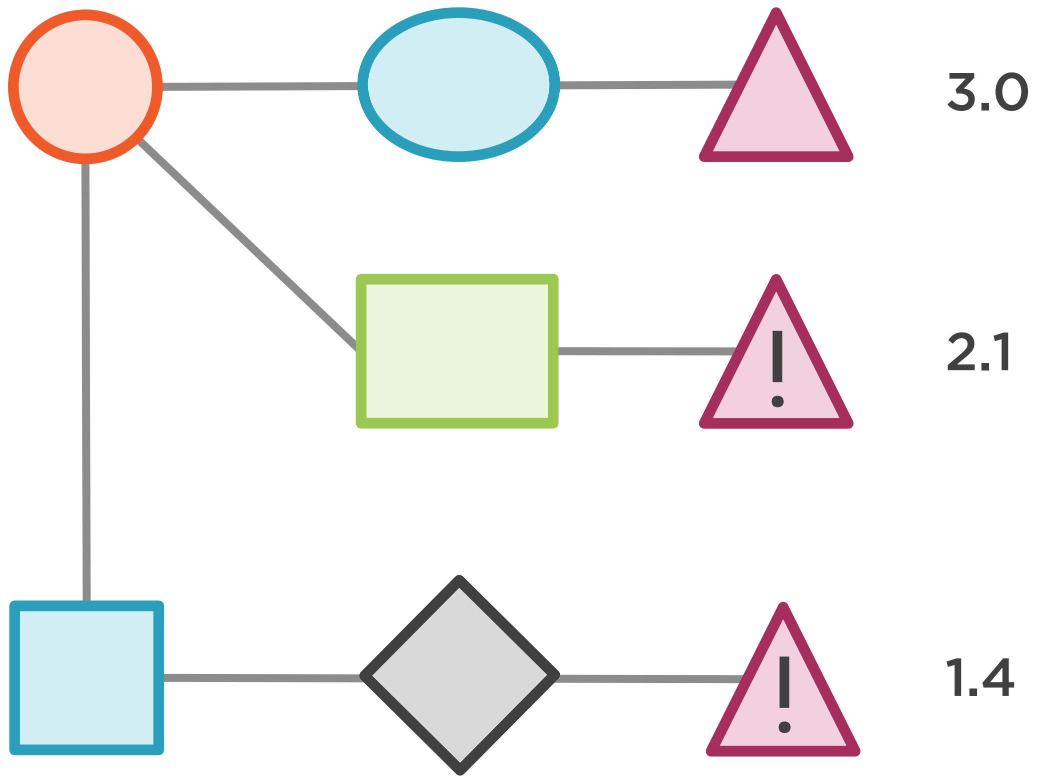
Creates duplicate commit in each branch
- Can cause confusion

Cherry-pick is an advanced feature
- Should not replace merge

Cherry-pick to Feature Branch



Cherry-pick to Version Branches



```
git log --oneline

git log <branch-name> --oneline
```

◂ Find the commit you want

```
git checkout <branch-name>
```

◂ Where do you want to put the commit? Checkout that branch.

```
git cherry-pick <commit>
```

◂ Perform the cherry-pick to append the commit to HEAD

Summary

Move code from branch to branch

Fast-forward merge

Rebase
- Squash commits
- Linear changes from another branch

Cherry-pick
- Move specific commits

Up Next:
Using Git Branches with Your Team