Scaling beyond one developer

Lessons learnt at Badoo

Daniel Thorpe - @danthorpe

Social network, with 180M users, for meeting new people and dating.

End of year, top-ten grossing app in the AppStore.

Company with big ambitions, wanting to work on more features and applications.

Hire!

Examine our process

Who uses SCM?

git?

GitHub?

Forks & Pull Requests?

```
$ git commit –a –m “Implemented all the stuff!!1”

$ git push origin master
```

But GitHub's awesome? Right?

Lacks structure for moving work from an idea to release.

GitHub limits what can be readily achieved regarding the development of release tools.

Continuous integration, QA, deployment, project management etc.

We use gitosis* and JIRA**

http://git-scm.com/book/ch4-7.html

* it doesn’t matter what you use

** it still doesn’t matter

Features

Features are well specified

Features have tickets

Developers work on features

Developers work on branches

Each ticket has a branch, a *feature branch*

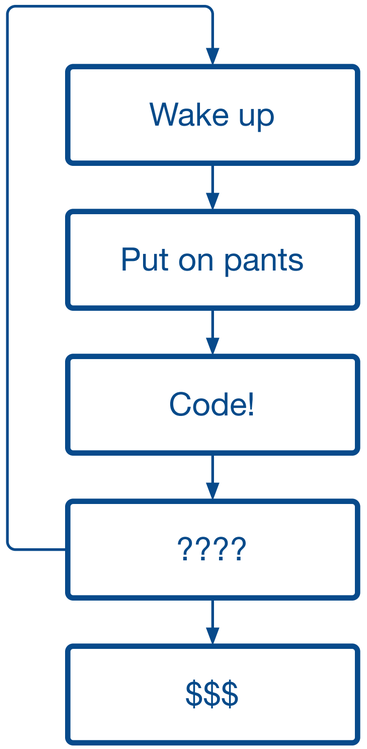
Feature branches

Bah! Pull Requests **are** features!

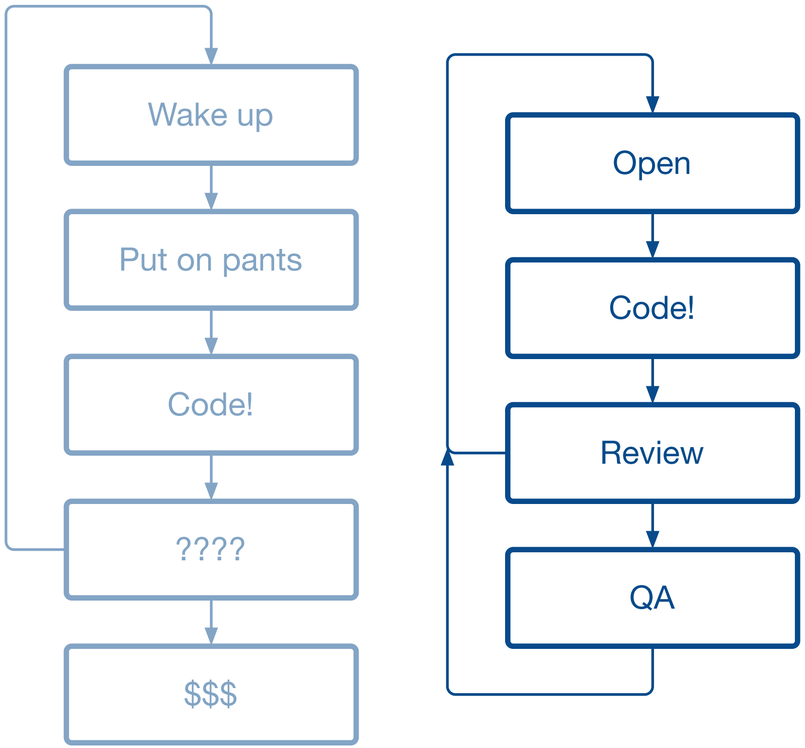
Feature branches grant a *well defined process* of development, code review, quality assurance and integration.

Allow large tasks to be broken up for multiple developers to work on at the same time.

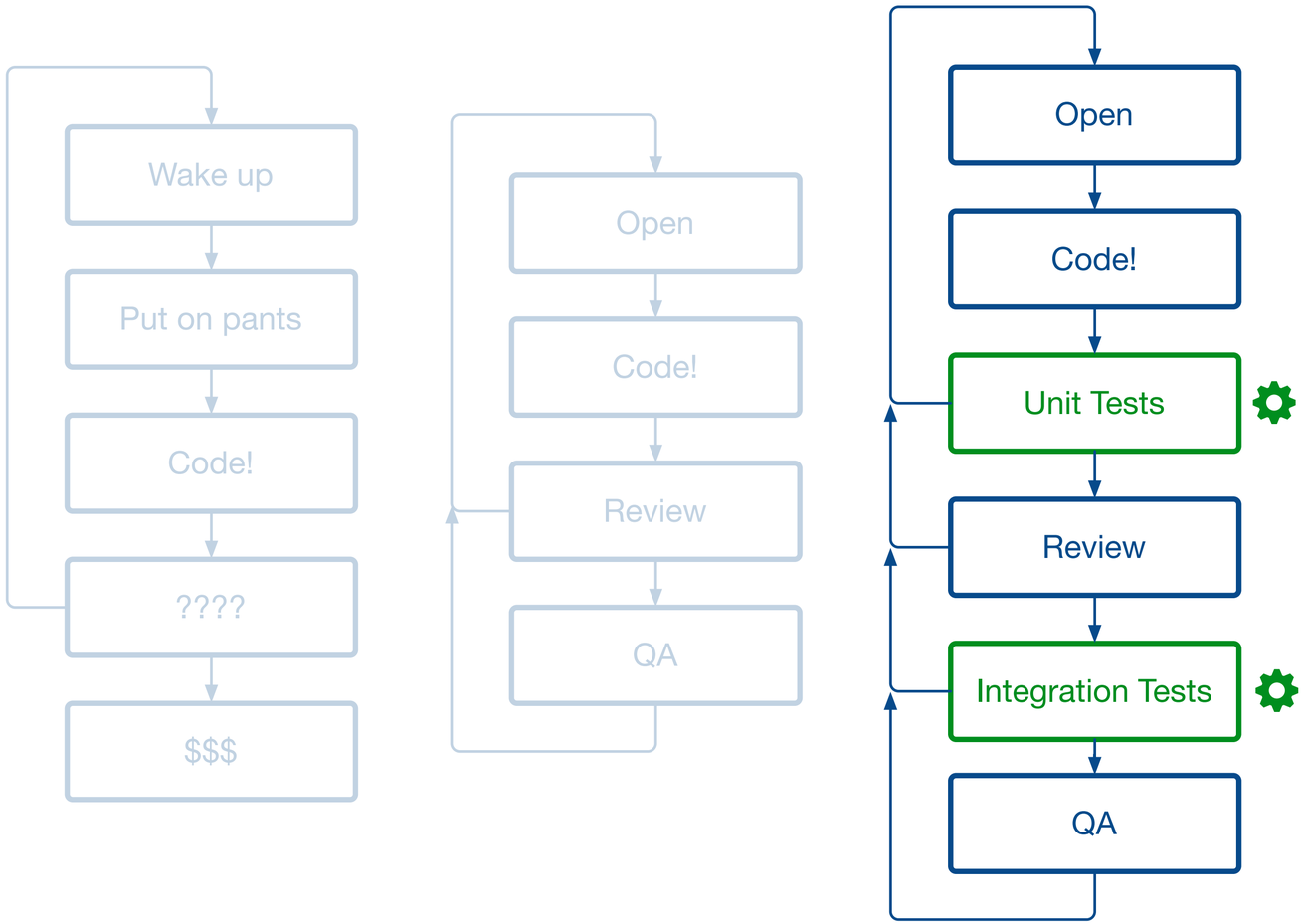
Branch Based Development



Branch Based Development



Branch Based Development



Automate everything

Create branches with a script

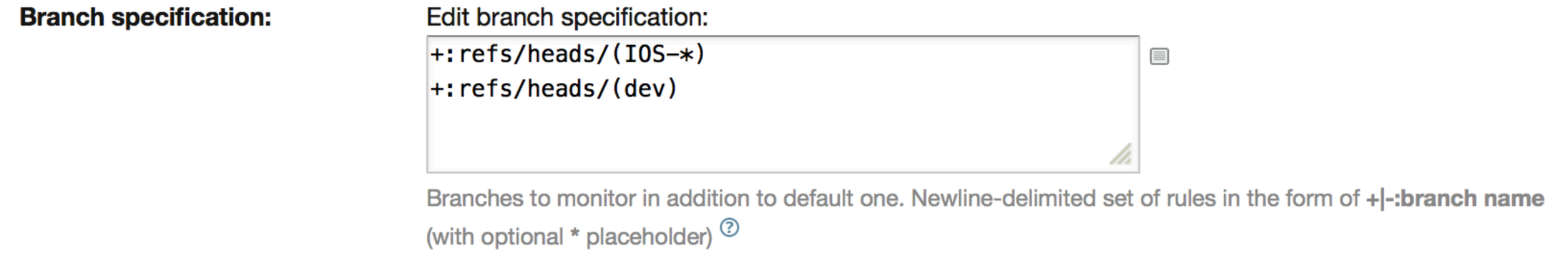
```
$ ./aida
$ Enter issue number
$ IOS-1234
$ Remote branch created, execute ‘git fetch origin; git checkout -b IOS-1234_Change_registration_screen’
```

Branches have a consistent naming convention

```
$ git branch -r
   origin/IOS-1023_Anonymous_Chat_Polish
   origin/IOS-102_Application_crashes_when_I_try_to_summon_map_Chat_and_Fake_location
   origin/IOS-1030_interestsCategories_leadToWrongPlace
   origin/IOS-1032_iPhone_iPad_Take_Photo_and_Choose_From_Existing_mixed_and_Take_Photo
   origin/IOS-1035_Fix_the_issue_with_non_thread_safe_animations_in_BMAChatFlowViewController
```

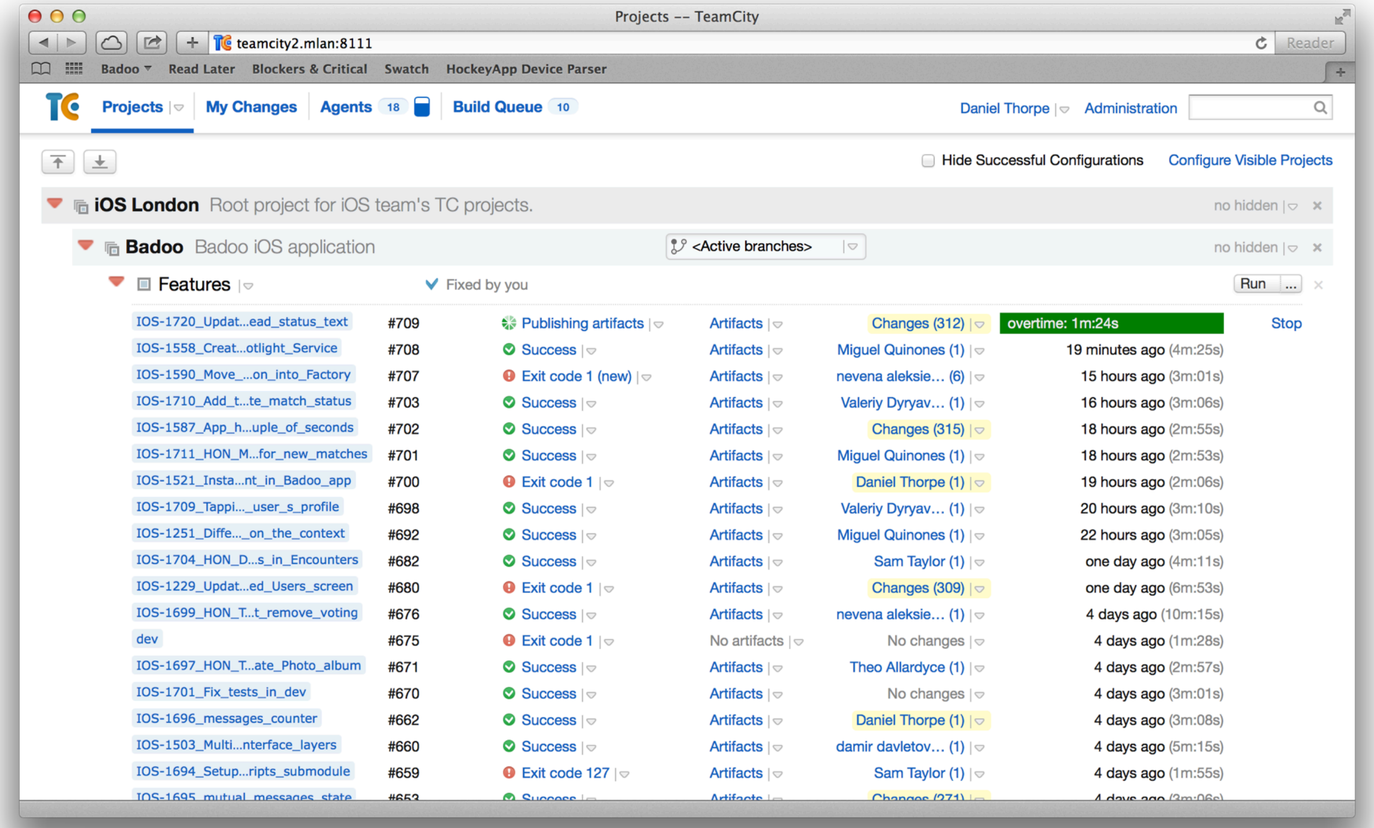
Automate everything

Configure continuous integration to build branches using regular expressions

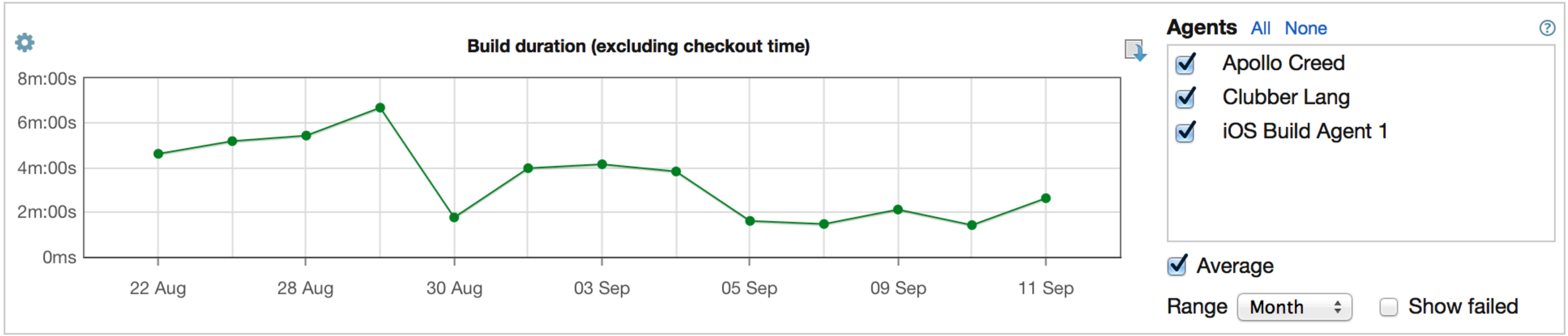


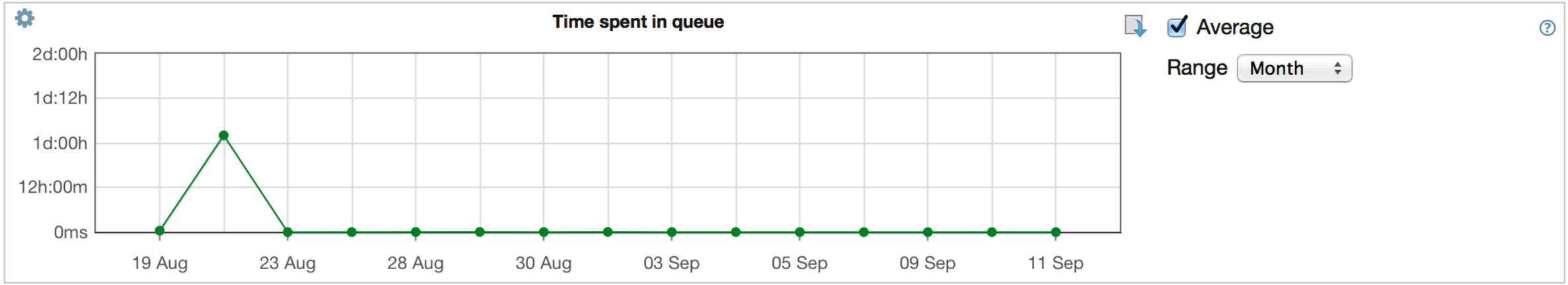
Automate everything

Configure continuous integration to build branches using regular expressions



Automate everything





Automate everything

Use a post-commit hook to prepend commit message logs

```
$ git status
# On branch IOS-1603_HON_Encounters_info_panel
nothing to commit, working directory clean

$ git log --no-merges --pretty=oneline
e45fc6a8cc [IOS-1603]: Adds Info Panel to Profile Pages.
34bc17b301 [IOS-1603]: WIP fixing the positioning of the toolbar in encounters.
dc05fc765e [IOS-1603]: Re-orders the protocol definitions
```

Use tab-to-complete in bash for local & remote branch names

```
$ curl https://raw.github.com/git/git/master/contrib/completion/git-completion.bash -o ~/.git-
completion.bash

if [ -f ~/.git-completion.bash ]; then
  . ~/.git-completion.bash
fi
```

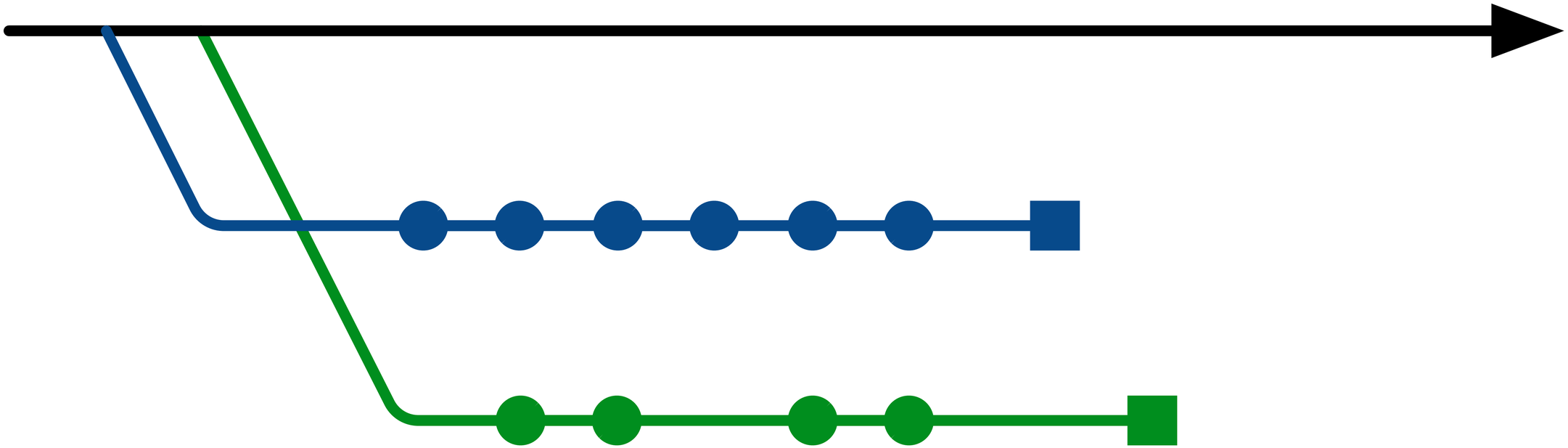
Integrating Features

Eventually features need releasing

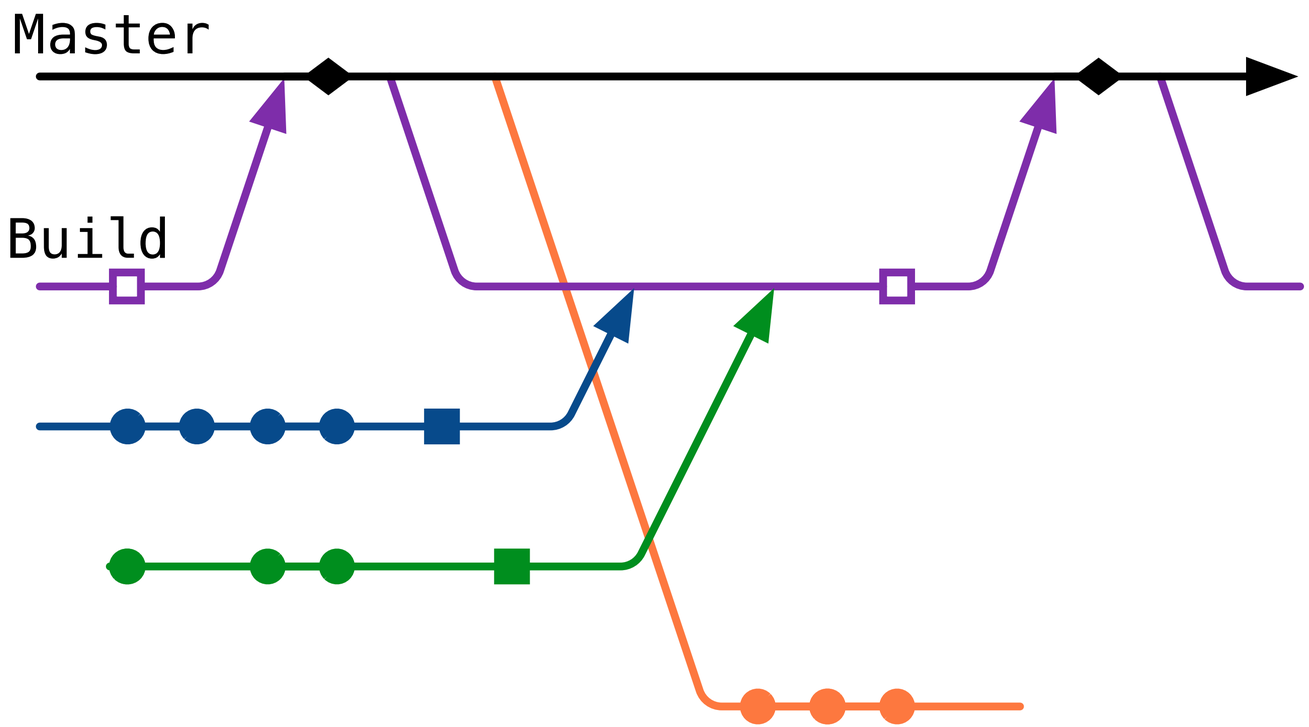
Feature branches need to be merged

But how? I'm too busy writing code...

Merging Feature Branches



The Web Way



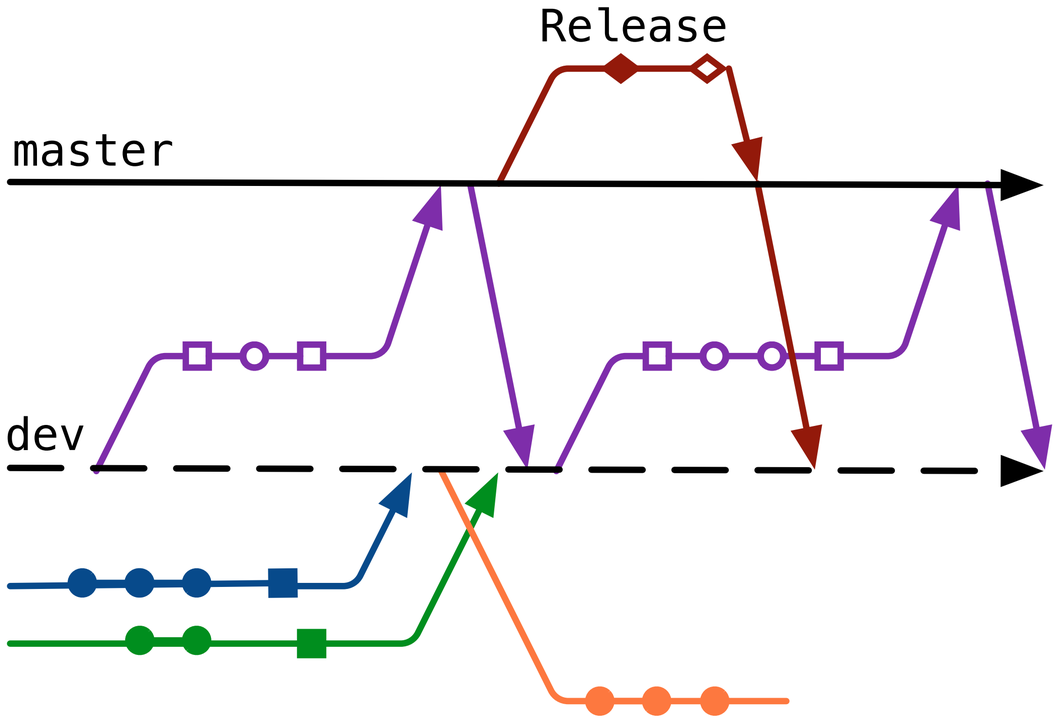
iOS doesn’t release twice per day

Too Slow

Developers get blocked by each other and perhaps themselves!

Process is strongly dependent on QA resources

origin/dev



Trade offs

Developers can safely share branches much quicker

More pragmatic - iOS doesn’t need continuous deployment

Easier to break large work up into a many of smaller tickets

Requires run-time feature flags

Once tickets are in dev - cannot be removed

Feature Flags

Programatic interface to disable functionality at run time on the client side

Only needed during the development of a whole feature

Allows for partial implementation of features to go into master

Scaling further

Encapsulate application features as re-usable modules

Use CocoaPods, instead of git submodules.

Increase code reuse via a Platform, distributes as a private CocoaPod

Use themes, styles, appearance to customise UI

Developing our own internal deployment tools to distribute feature builds internally

Provide designers and product owners with a UI catalog to further increase component reuse

Questions?

Daniel Thorpe - @danthorpe