Git and GitHub Workflows:

Agile Branching Strategies
structure / waterfall
Plan
Design
Develop
Deploy
chaos? / agile
flexibility / agile
iterations / agile
Sketch
Prototype
Review
Refine
README driven development
Sketch...
A simple and elegant wireframe kit.
Light weight, clear and exactly what you need.
Sqetch, an Illustrator Wireframe Toolkit
Prototype...
Sample example-maven repo for GitHub Training classes https://training.github.com — Edit

- 21 commits
- 1 branch
- 0 releases
- 2 contributors

Branch: master / example-maven /

Pruning this down to Maven-only

- resources: Addition of the README and basic Groovy source samples.
  - 3 years ago
- src: Adding maven build script
  - 2 years ago
- .gitattributes: .gitattributes to make this play nicely on Windows
  - 2 years ago
- .gitignore: Adding maven build script
  - 2 years ago
- .travis.yml: Travis CI
  - 2 years ago
- README.txt: Even I can change the readme file
  - 11 months ago
- pom.xml: Adding maven build script
  - 2 years ago
- runme.sh: Added shell script to drive the execution of the app
  - 3 years ago

Cloning...
HTML prototypes even for native apps
**Atom Shell**

The Atom Shell framework lets you write cross-platform desktop applications using JavaScript, HTML and CSS. It is based on node.js and Chromium and is used in the Atom editor.

**Downloads**

Prebuilt binaries of atom-shell for Linux, Windows and Mac can be found on the [releases](https://github.com/atom/atom-shell/releases) page.

**Documentation**

Guides and the API reference is located in the [docs](https://github.com/atom/atom-shell/blob/master/docs) directory. It also contains documents describing how to build and contribute to atom-shell.
Review...
The Hubot Lab  Where we make Hubot do things and stuff

Matthew McCullough
/what's undeployed?

Hubot
Here's what's not deployed on master yet: site/not_deployed

Matthew McCullough
/help

Hubot
View paste

hubot good news everyone! <news> - Generates Professor Farnsworth
hubot help (statuslist) - Help status.
hubot help url.
hubot if <text>, <word that can start a question> <text>? - Generates Philosoraptor
hubot khanify <text> - TEEEEEEEEEEEEEXT!
hubot one does not simply <text> - Lord of the Rings Boromir
hubot pa - The latest Penny Arcade comic
hubot pa <date> - Penny Arcade comic from <date> - mm/dd/yyyy format
hubot remember <location> as a <group> location - Remembers the location for the group
hubot remind me in <time> to <action> - Set a reminder in <time> to do an <action> <time> is in the format 1 day, 2 hours, 5 minutes et
hubot where can we go for <group>? - Returns a list of places that exist for the group
hubot where should we go for <group>? - Returns a randomly selected location for the group
hubot wiki me <query> - Searches for <query> on Wikipedia.
i verb you - Get a relevant response back. i am 'maddox' - map your user to the github login 'maddox'
image me <query> - The Original. Quartes Google Images for <query> and
in <time> <command> - Perform a command after some time passes.
index user - index the "user"
tclcm commented on pull request github/projects#5
Refine...
Phil Haack
@KentBeck and if nothing changes.

Kent Beck
If you don’t make mistakes and you have perfect knowledge, you don’t have to iterate.
How we use Pull Requests to build GitHub

We recently shipped a new About section. It has all sorts of stuff like high resolution logos, pictures of the GitHub team, a little bit about our story, recent press mentions and maybe most importantly positions we’re hiring for. It’s awesome.

But that’s not the point of this post. Instead, let’s take a look at how we used a massive Pull Request to ship this feature.

We talk a lot about how GitHub works in blog posts and talks and this is a great example of how we use Pull Requests.

Here is what the PR looked like for the new About page:
130 commits
91 comments
45 deployments
10 contributors
8 weeks
Probably the most important technical book of 2010

-Martin Fowler, ThoughtWorks
Distributed VCS
Spatialization is the result of the ForceAtlas layout, implemented in the open source graph visualization and manipulation project easy-designs.
many clones
many repos
many contributors
Central Repo

- Alice
- Bob
- Cory
- Dana
Alice → Local Repo → Central Repo

Bob → Local Repo → Central Repo

Cory

Dana
many forks
<table>
<thead>
<tr>
<th>Author</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tim Carey-Smith</td>
<td>c6774ad939f90b3d7dff9e9eb7048a79df4a457ec</td>
</tr>
<tr>
<td></td>
<td>Allow the status command to accept a task or group name.</td>
</tr>
</tbody>
</table>
many branches
master branch

topic branches
many merges
b4f86a4 Git 2.0-rc2
* 35936f8 Git 2.0-rc1
* cc29195 Git 2.0-rc0
  * 68773ac Sync with 1.9.2
|\ | |
|* | 0bc85ab Git 1.9.2
| | 5172cb3 Sync with 1.9.1
|\ | |
* cee0c27 Git 1.9.1
  * e3f1185 Merge branch 'cc/starts-n-ends-with-endgame'
  |
* 666c90b strbuf: remove prefixcmp() and suffixcmp()
  |
* 2687ffd Merge branch 'jc/hold-diff-remove-q-synonym-for-no-deletion'
  |
* c48f681 diff: remove "diff-files -q" in a version of Git in a distant future
  |
* 289ca27 Merge branch 'gj/push-more-verbose-advice'
  |
* 11037ee push: switch default from "matching" to "simple"
  |
* 2b4a888 Merge branch 'jc/core-checkstat-2.0'
  |
* c1b5d73 core.statinfo: remove as promised in Git 2.0
  |
* 160c4b1 Merge branch 'jc/add-2.0-ignore-removal'
  |
* fdc97ab git add <pathspec>... defaults to "-A"
Meet Jenkins
Find out what Jenkins is and get started.

Use Jenkins
See how to get more out of your Jenkins.
hubot
The Hubot Lab Where we make Hubot do things and stuff

matthewmccullough Matthew McCullough
/what's deployed?

Hubot Here's what's not deployed on master yet: site/not deployed

matthewmccullough Matthew McCullough
/help

Hubot View paste

hubot good news everyone! <news> - Generates Professor Farnsworth
hubot help (statuslist) - Help status.
hubot help url.
hubot if <text>, <word that can start a question> <text>? - Generates Philosoraptor
hubot khanify <text> - TEEEEEEEEEEEEEEEEEXT!
hubot one does not simply <text> - Lord of the Rings Boromir
hubot pa - The latest Penny Arcade comic
hubot pa <date> - Penny Arcade comic from <date> - mm/dd/yyyy format
hubot remember <location> as a <group> location - Remembers the location for the group
hubot remind me in <time> to <action> - Set a reminder in <time> to do an <action> <time> is in the format 1 day, 2 hours, 5 minutes et
hubot where can we go for <group>? - Returns a list of places that exist for the group
hubot where should we go for <group>? - Returns a randomly selected location for the group
hubot wiki me <query> - Searches for <query> on Wikipedia.
i <verb> you - Get a relevant response back.
i am 'maddock' - map your user to the github login 'maddock'
image me <query> - The Original. Quarters Google Images for <query> and
in <time> <command> - Perform a command after some time passes.
index user - index the "user"
git-flow
From this point on, "next release" means the release *after* 1.0

Start of release branch for 1.0

Only bugfixes!

Bugfixes from rel. branch may be continuously merged back into develop

Tag 1.0
Initialized empty Git repository in /Users/mccm06/Documents/Scratch/project1/.git/

Scratch/project1 git:(master) ➤ git flow

usage: git flow <subcommand>

Available subcommands are:

- init: Initialize a new git repo with support for the branching model.
- feature: Manage your feature branches.
- release: Manage your release branches.
- hotfix: Manage your hotfix branches.
- support: Manage your support branches.
- version: Shows version information.

Try 'git flow <subcommand> help' for details.
Switched to a new branch 'hotfix/heartbleed'

Summary of actions:
- A new branch 'hotfix/heartbleed' was created, based on 'master'
- You are now on branch 'hotfix/heartbleed'

Follow-up actions:
- Bump the version number now!
- Start committing your hot fixes
- When done, run:

  `git flow hotfix finish 'heartbleed'`

≡ Scratch/project1  git:(hotfix/heartbleed) ➤ git checkout master
Switched to branch 'master'
≡ Scratch/project1  git:(master) ➤ git flow hotfix finish heartbleed
Deleted branch hotfix/heartbleed (was 590ba50).

Summary of actions:
- Latest objects have been fetched from 'origin'
- Hotfix branch has been merged into 'master'
- The hotfix was tagged 'heartbleed'
- Hotfix branch has been back-merged into 'develop'
- Hotfix branch 'hotfix/heartbleed' has been deleted

≡ Scratch/project1  git:(master) ➤  
simplest thing that works
simplest thing that works
a long-lived release needs stability
embedded code needs stability
customized apps needs stability
simplest thing that works
GitHub flow
Pull request?
branch with identity
branch with conversation
branch with state
rebasing branches
testing demands
quick builds
quick tests
quick deployments
Feature Flag beta code
FeatureToggle

29 October 2010

One of the most common arguments in favor of FeatureBranch is that it provides a mechanism for pending features that take longer than a single release cycle. Imagine you are releasing into production every two weeks, but need to build a feature that's going to take three months to complete. How do you use Continuous Integration to keep everyone working on the project?
One of the most common arguments in favor of FeatureBranch is that it provides a mechanism for pending features that take longer than a single release cycle. Imagine you are releasing into production every two weeks, but need to build a feature that's going to take three months to complete. How do you use Continuous Integration to keep everyone working on the mainline without revealing a half-implemented feature on your releases? We run into this issue quite a lot and feature toggles are a handy tool to deal with it.
Embrace smaller iterations
focused ideas
focused changes
focused sketches
focused prototypes
Git and GitHub Workflows:
Agile Branching Strategies
Q&A