Agile-ish life at GitHub

at Turku Agile Day
GitHub
agile what?
agile organizations
agile teams
agile development
agile features?
waterfall...
is software ever done?
is software ever perfect?
is code ever perfect?
is code ever bug free?
four years of changes
32 colleagues

2 teams
50 colleagues

2 teams
75 colleagues

3 teams
110 colleagues
4 teams
150 colleagues

5 teams
200 colleagues

10 teams
220 colleagues

12 teams
240 colleagues

15 teams
13 leaders
13 managers?
deploy all the time
values creativity
The difference between machines and human beings is that human beings can be reproduced by unskilled labor.

— Arthur C. Clarke
A CUSTOMIZABLE, KEGERATOR-POWERED LIFE EMBETTERMENT ROBOT

COMMISSIONED BY GITHUB

☐ I WOULD LIKE TO VIEW HUBOT'S SOURCE CODE
(OBTAINABLE AT HTTP://GITHUB.COM/GITHUB/HUBOT/)

☐ I WOULD LIKE TO OBTAIN MY OWN HUBOT
(VERSION 2.2.0)
Coordinate
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.
Sqeetch, 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

- matthewmccullough authored 9 months ago
  - latest commit 59f3330105

- 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

- README.txt
HTML prototypes even for native apps
Cross-platform desktop application shell — Edit

1,530 commits 1 branch 67 releases 0 contributors

branch: master atom-shell /

Merge pull request #243 from hughsk/enable-webgl

<table>
<thead>
<tr>
<th>Author</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>zcbenz</td>
<td>authored 3 hours ago</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>File</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>atom</td>
<td>Enable WebGL</td>
</tr>
<tr>
<td>chrome</td>
<td>Add LICENSE in chroma/</td>
</tr>
<tr>
<td>docs</td>
<td>Fix subtitle</td>
</tr>
<tr>
<td>script</td>
<td>win: Fix dumping symbols.</td>
</tr>
<tr>
<td>spec</td>
<td>Listen on 0 to get random port</td>
</tr>
<tr>
<td>tools</td>
<td>win: Fix dumping symbols.</td>
</tr>
<tr>
<td>vendor</td>
<td>Install npm from npm, fixes #236.</td>
</tr>
<tr>
<td>.gitignore</td>
<td>Add vendor/npm to .gitignore.</td>
</tr>
<tr>
<td>gitmodules</td>
<td>Use tabs instead of spaces</td>
</tr>
<tr>
<td>.travis.yml</td>
<td>Add initial Travis config</td>
</tr>
<tr>
<td>LICENSE</td>
<td>Switch to use MIT license.</td>
</tr>
</tbody>
</table>

You can clone with HTTPS, SSH, or Subversion.

Clone in Desktop
Download ZIP
Atom Shell  build passing

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, Window and Mac can be found on the releases page.

Documentation

Guides and the API reference is located in the docs directory. It also contains documents describing how to build and contribute to atom-shell.
Review...
tclem 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
many clones
many repos
many contributors
many forks
<table>
<thead>
<tr>
<th>User</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>mojombo</td>
<td></td>
</tr>
<tr>
<td>Bertg</td>
<td></td>
</tr>
<tr>
<td>kevinclark</td>
<td></td>
</tr>
<tr>
<td>raggi</td>
<td></td>
</tr>
<tr>
<td>halorgium</td>
<td></td>
</tr>
</tbody>
</table>

Tim Carey-Smith

c6774ad939f90b3d7dff9eeb7048a79df4a457ec

Allow the status command to accept a task or group name.
many 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"
Welcome to open source Janky, our continuous integration server. It's built on Hubot, and GitHub.

Meet Jenkins
Find out what Jenkins is and get started.

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

matthewmccullough Matthew McCullough
/what's undeployed?

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

matthewmccullough Matthew McCullough
/help

Hubot
View paste

hubot good news everyone! <news> - Generates Professor Farnsworth
hubot help (statuslist) - Help status.
hubot help url - Help url.
hubot if <text>, <word that can start a question> <text>? - Generates Philosoraptor
hubot khanify <text> - TEEEEEEEEEEXT!
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 - Searches for <query> on Wikipedia. i <verb> you - Get a relevant response back.

i'm maddox - Map your user to the github login 'maddox'
imagine me <query> - The Original. Quirits 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/

```bash
$ cd 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:

```bash
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
3D5 → 671 → A92' → 48E'

3D5
- js/app.js
- placeholder

671
- js/app.js
- index.html

A92'
- js/view/user.js
- team.jpg

48E'
- js/model/user.js
- about.html

feature

master
testing demands
quick builds
quick tests
quick deployments
feature flag
FeatureToggle

29 October 2010
tags: continuous integration

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 development branch?
FeatureToggle

29 October 2010
tag: continuous integration

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
focused deploys
“Remember, always, that everything you know, and everything everyone knows, is only a model. Get your model out there where it can be viewed. Invite others to challenge your assumptions and add their own.”

– Donella H. Meadows, Thinking in Systems: A Primer
Agile-ish life at GitHub