Git Branching for Continuous Delivery
not afraid to ham it up in photos

Sarah Goff-Dupont
Developer Advocate & Agile Delivery Nerd

@DevToolSuperFan
Agenda

1. Git + CD = BFFs
2. Branching models
3. Pointers & pro-tips
Why Git for CD?
Challenge #1: dependencies

easy to navigate, right?
Challenge #2: code line chaos

I do my thing, too

I do my thing

look Ma, a goat!
Build in narrow vertical slices

"big bang" launch

pint-sized release
In Git

Branching & merging is a breeze
Keep the main line clean

dev branch = isolation chamber
<table>
<thead>
<tr>
<th>Author</th>
<th>Commit</th>
<th>Message</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stacey Shikura</td>
<td>81b5989</td>
<td>Forgot to change method signature</td>
<td>48 minutes ago</td>
</tr>
<tr>
<td>Stacey Shikura</td>
<td>631444d</td>
<td>Moved bigger</td>
<td>48 minutes ago</td>
</tr>
<tr>
<td>Eli Bishop</td>
<td>0d570c-f</td>
<td>AMKT-6543: comments</td>
<td>57 minutes ago</td>
</tr>
<tr>
<td>Eli Bishop</td>
<td>6f8e512</td>
<td>Merge branch <code>eb/AMKT-memcached</code> into ebi/AMKT-6543</td>
<td>an hour ago</td>
</tr>
<tr>
<td>Eli Bishop</td>
<td>6f5c54c</td>
<td>AMKT-6543: ensure that we don't put unserializable things into FlashMap</td>
<td>an hour ago</td>
</tr>
<tr>
<td>Eli Bishop</td>
<td>2b66e0</td>
<td>Merged in ebi/AMKT-6522 (pull request #1215)</td>
<td>3 hours ago</td>
</tr>
<tr>
<td>Stacey Shikura</td>
<td>c88d78</td>
<td>Review comments</td>
<td>20 hours ago</td>
</tr>
<tr>
<td>Arun Bhalla</td>
<td>e644c7f</td>
<td>Merged in arun/AMKT-6566 (pull request #1221)</td>
<td>22 hours ago</td>
</tr>
<tr>
<td>Eli Bishop</td>
<td>94b654b</td>
<td>Merged in ebi/AMKT-6579 (pull request #1227)</td>
<td>22 hours ago</td>
</tr>
<tr>
<td>Arun Bhalla</td>
<td>72d5852</td>
<td>Merged in arun/AMKT-6577 (pull request #1228)</td>
<td>22 hours ago</td>
</tr>
<tr>
<td>Ryan Talusan</td>
<td>92f0a8e</td>
<td>Merged in rt/AMKT-6568 (pull request #1225)</td>
<td>23 hours ago</td>
</tr>
<tr>
<td>Stacey Shikura</td>
<td>580f250</td>
<td>fixing branch</td>
<td>yesterday</td>
</tr>
<tr>
<td>Ryan Talusan</td>
<td>1e7b1e9</td>
<td>AMKT-6588 - apply code review comment</td>
<td>yesterday</td>
</tr>
</tbody>
</table>
Branch-per-Issue Workflow
A branch for every issue

fortunately, no goats here
A branch for every issue
A branch for every issue

still no goats!
Surprise!
Using an integration branch
Using an integration branch

![Diagram showing the integration branch flow with branches sgd-IRKD-30, sgd-IRKD-45, integration, and master.](image-url)
Branch-per-Issue Workflow

for installed app teams
Multiple-version support

v 1.1

v 1.2

master

sgd-IRKD-30
Pre-release changes
Post-release updates

- v 1.2
- master
- bugfix-IRKD-32
- sgd-IRKD-30
Post-release updates

- v 1.2
- v 1.1
- master
- bugfix-IRKD-32
Other "Stuff" to Consider
Running CI on dev branches

all active branches are under test
Running CI on dev branches

1. Manually clone master's CI configs
2. Automate with plugin or Git hook

bitbucket.org/tpettersen/git-ci-hooks
Running CI on dev branches
Using build power efficiently

push-button builds here

trigger CI with every commit here

sgd-IRKD-30

v 1.2

master
Peer code review

PLEASE SIR
I WANT SOME MERGE
Peer code review

1. Create request via UI or `git request-pull`
2. Review, revise, rinse & repeat
3. Approve & merge
Take it to the extreme

1. Build small pieces, end-to-end
2. Let dev branches do the dirty work
3. Keep master releasable at all times
4. Reach continuous delivery nirvana
More info at...

1. http://atlassian.com/git
2. @Atlassian
Q & A
Evaluate This Session

1. Sign-in: www.eclipsecon.org

2. Select session from schedule

3. Evaluate: +1 0 -1
Sarah Goff-Dupont

Developer Advocate & Agile Delivery Nerd

@DevToolSuperFan
sgoffdupont@atlassian.com