



AUTOMATING QA

in Drupal

Or “Why you should be testing”

INTRODUCTIONS

- Dave Stoline

Acquia, Cloud Engineering Manager for DevTools

- Matthew Grasmick

Acquia, Technical Team Lead

MATT HAD A BABY!

- Clearly, I'm not two people :)

POLL

Who here has written a test for a Drupal site?

(e.g., simpletest, load test, Behat feature, unit test)

- For specific site (not core, not contrib)?
 - E.g., testing implementation of Blog for something.gov.
 - This is our focus today.

Who here uses automated testing as part of their every day development workflow?

MOST DRUPAL PROJECTS
DON'T USE AUTOMATED QA

THIS IS A PROBLEM

Quality assurance matters, and it doesn't happen without regular and consistent testing.

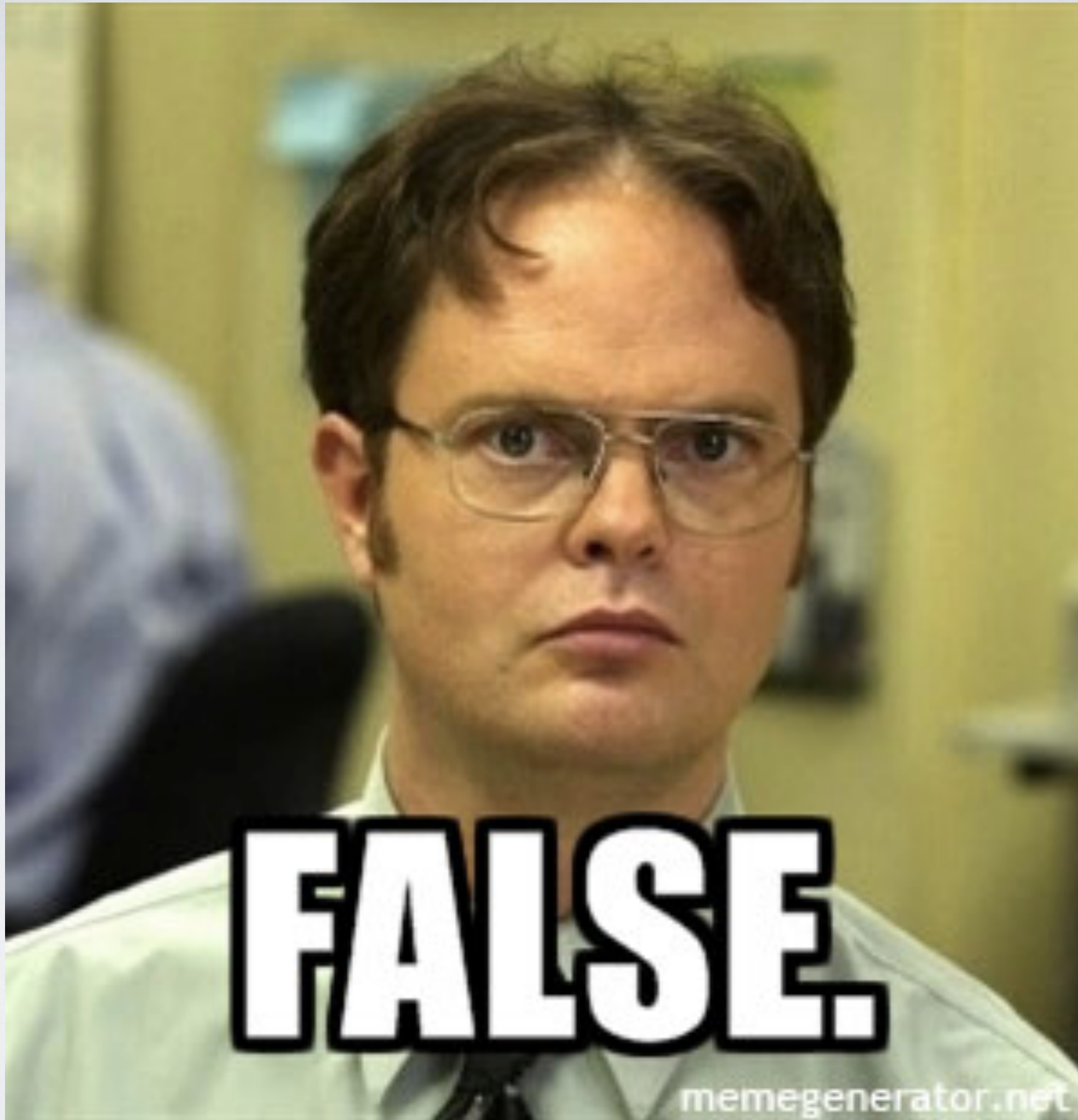
THE (SAD) STATE OF QA IN DRUPAL

... for most Drupal projects ...

- **QA is vaguely defined.** Examples you may see on a contract:
 - “QA will be performed.”
 - “x hours will be spent testing the site.”
 - “Site will be fully functional in IE 9+, Chrome, FireFox, and Safari.”
- **QA is inconsistently performed.** Examples of real QA practices:
 - “Everybody please click around and tell me if you see any bugs.” - Real thing.
 - QA team verifies that “everything is working.”

HOW DID WE GET HERE?

- “We don’t have the budget for testing.”
- “If we spend time testing, we won’t hit our deadline.”
- “We don’t have the expertise or infrastructure.”



QA IS UNDERVALUED

It saves time, money, and *improves quality*.

“Quality is the ally of schedule and cost, not their adversary. If we have to sacrifice quality to meet schedule, it’s because we are doing the job wrong from the very beginning.”

--James A. Ward

“The bitterness of poor quality remains long after the sweetness of meeting the schedule has been forgotten.”

--Karl Wiegers

QA PAYS OFF

(AUTOMATED)

- Lowers cost over project lifetime.
 - Improves development efficiency.
 - Catches bugs earlier, reducing necessity for refactoring.
 - Reduces overhead of feedback loop btw devs & testers.
 - Speeds up release and deployment process.
- Improves the product.
 - Users don't like bugs.
 - Execs don't like bugs.
- Improves code quality and maintainability.
 - Code is more readable and consistent.
- Reduces the risk of regression.
- Catches edge cases.

MOVE FAST

- Want to upgrade to Drupal 8 today or make other gutsy major refactorings?

**HAVING SOLID QA WILL ENABLE
YOU TO EXECUTE FASTER**

“... defect density of the four products decreased between 40% and 90% relative to similar projects that did not use the TDD practice.”

—Study performed by Microsoft

I'M SOLD! WHERE DO I START?

NOT WITH MANUAL TESTING

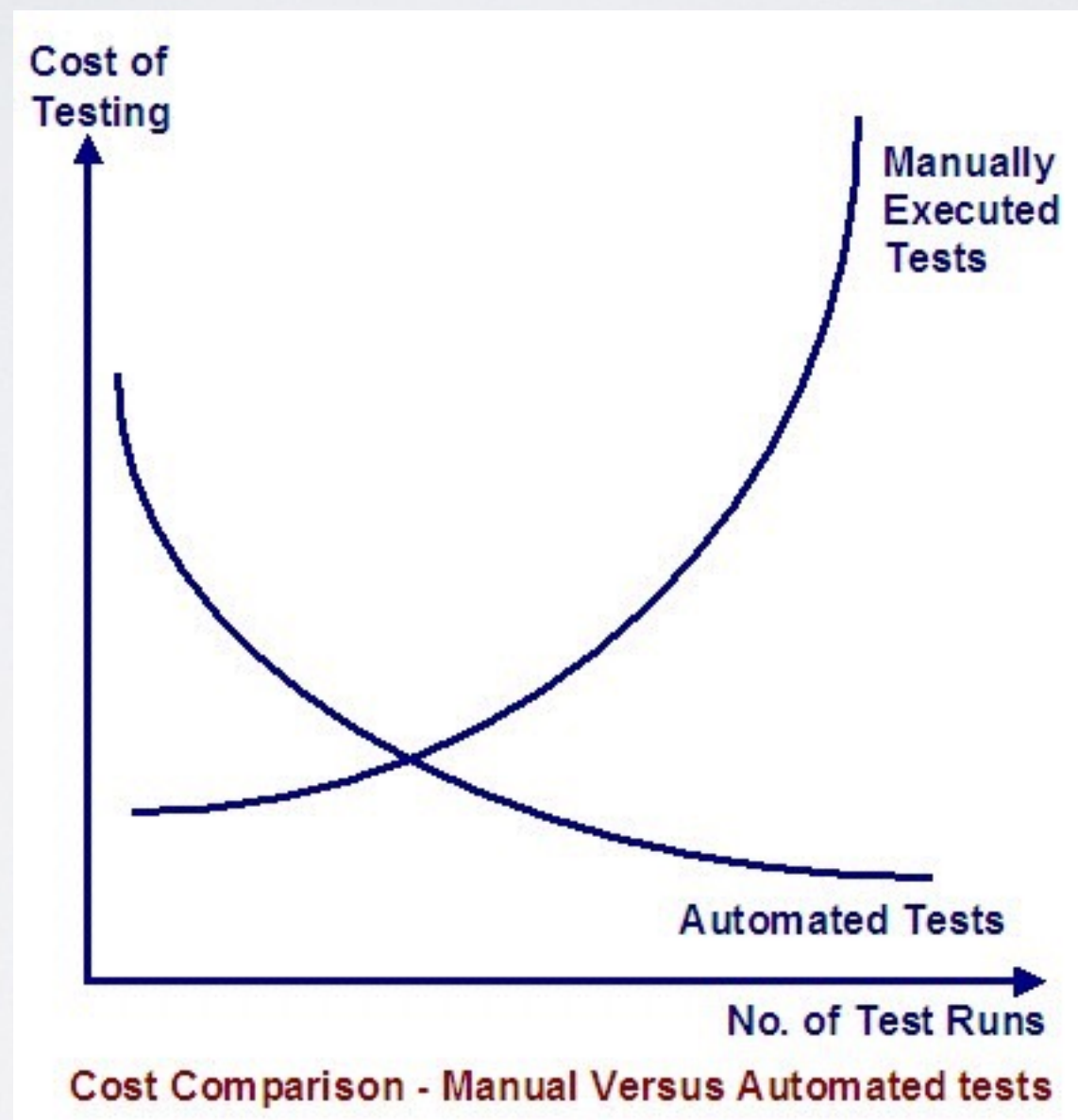
MANUAL TESTING

- Quality depends on the individual tester, often leading to inconsistent standards and quality
- Speed of testing depends on individual tester
- As the application grows in size, manual testing becomes more costly
- Manual testing often is not transparent.
 - No standard record or log of tests performed
 - Often difficult to reproduce issues

THE SOLUTION?

Robots.

YES, ROBOTS!



WHAT ARE ROBOTS GOOD AT?

- Regression testing
- Code quality checks: (linting and sniffing)
- *Load testing*
- *Unit testing*

WHERE DO I GET ROBOTS?

- Let's look at specific solutions for:
 - Regression testing
 - Code Quality
- Then, we'll tie it into your every day life with continuous integration.

REGRESSION TESTING

Test that features are working, and make sure they keep working!

Examples:

Blog feature:

- Admin can create a blog node.
- The node displays the title, date, body, etc.
- The Blog view displays a teaser
- The RSS feed is updated

Search feature:

- There is a search box on the home page.
- When you enter “Help me!” and click “Search”, you are shown a list of search results.
- The results contain the work “Help”

SUGGESTED TOOL: **BEHAT**

- Behavior-Driven Development focus.
 - Tests are described as a set of user behaviors.
 - Tests are easy for everybody (even non-techies) to understand.
- PHP Based testing solution (built on Symfony components).
- Has Drupal integration OOTB.

REAL BEHAT TEST

@api @og @permissions

Feature: Organizations

Organizations are organic groups, configured to provide groups landing pages and segmented access control.

Scenario: Group administrator can administer group menus and group members

Given I am logged in as a user with the "**authenticated user**" role

And I am viewing an "**organization**" node

And I am an "**administrator**" of the current group

When I click "**Group**"

Then I should see the following <links>

	links	
	Add people	
	People	
	Menus	

EXAMPLE TEST RUN

@api @og @permissions

Feature: Organizations

Organizations are organic groups, configured to provide groups landing pages and segmented access control.

Scenario: Group administrator can administer group menus and group members

Given I am logged in as a user with the "authenticated user" role

And I am viewing an "organization" node

And I am an "administrator" of the current group

When I click "Group"

Then I should see the following <links>

links	
Add people	
People	
Menus	

1 scenario (1 passed)

5 steps (5 passed)

1m5.587s

HOW DOES THAT WORK?

- The test-language is parsed and paired with PHP methods via regexs.
- This is done by Behat automatically, don't be scared.
- You can customize the available “steps”

EXAMPLE STEP DEFINITION

```
/**
 * @Given /^I am (?:a|an) "([^"]*)" of the current group$/
 */
public function iAmAnOfTheCurrentGroup($group_role) {
    $group = $this->getCurrentGroupFromURL();
    $this->addMemberToGroup($group_role, $group);
    $this->getMainContext()->softReload();
}
```


INTEGRATE INTO WORKFLOW!

- Product owners can write user story in test language
- Devs can use the user story as the test
- QA Analysts can see exactly which requirement is failing to be met

MORE ABOUT BEHAT:

behat.org

drupal.org/project/drupalextension

drupal.org/project/behattrunner

CODE QUALITY CHECKS

- Drupal Code Sniffer
 - Ensure that Drupal coding standards are being met
- PHP Linter
 - Verifies that PHP syntax is correct and can be run by PHP process
- Git pre-commit hooks
 - Enforce code quality standards (and other things) before code is ever committed

DRUPAL CODE SNIFFER (CODER)

- Project page:
<https://www.drupal.org/project/coder>
- Installation instructions:
<https://www.drupal.org/node/1419988>
- Example installation:

```
$ drush pm-download coder --destination=$HOME/.drush  
$ drush cache-clear drush  
$ drush install-php-code-sniffer
```


RUNNING CODE SNIFFER

```
$ drush drupalcs example.module
```

```
FILE: /home/klausl/workspace/coder/example.module
```

```
FOUND 5 ERRORS AFFECTING 5 LINES
```

```
1 | ERROR | [ ] Missing file doc comment
3 | ERROR | [ ] Missing function doc comment
4 | ERROR | [ ] Opening brace should be on the same line as the declaration
5 | ERROR | [x] Line indented incorrectly; expected 2 spaces, found 1
6 | ERROR | [x] Line indented incorrectly; expected 2 spaces, found 3
```

```
PHPCBF CAN FIX THE 2 MARKED SNIFF VIOLATIONS AUTOMATICALLY
```

PHP LINTER

- OOTB with PHP.
- Example lint check:

```
$ php -l example.module
```

USE GIT PRE-COMMIT HOOK

- Automatically runs drupalcs and linter every time you make a commit
- Pre-commit script:
<https://github.com/geraldvillorente/drupal-pre-commit>
- Example installation:

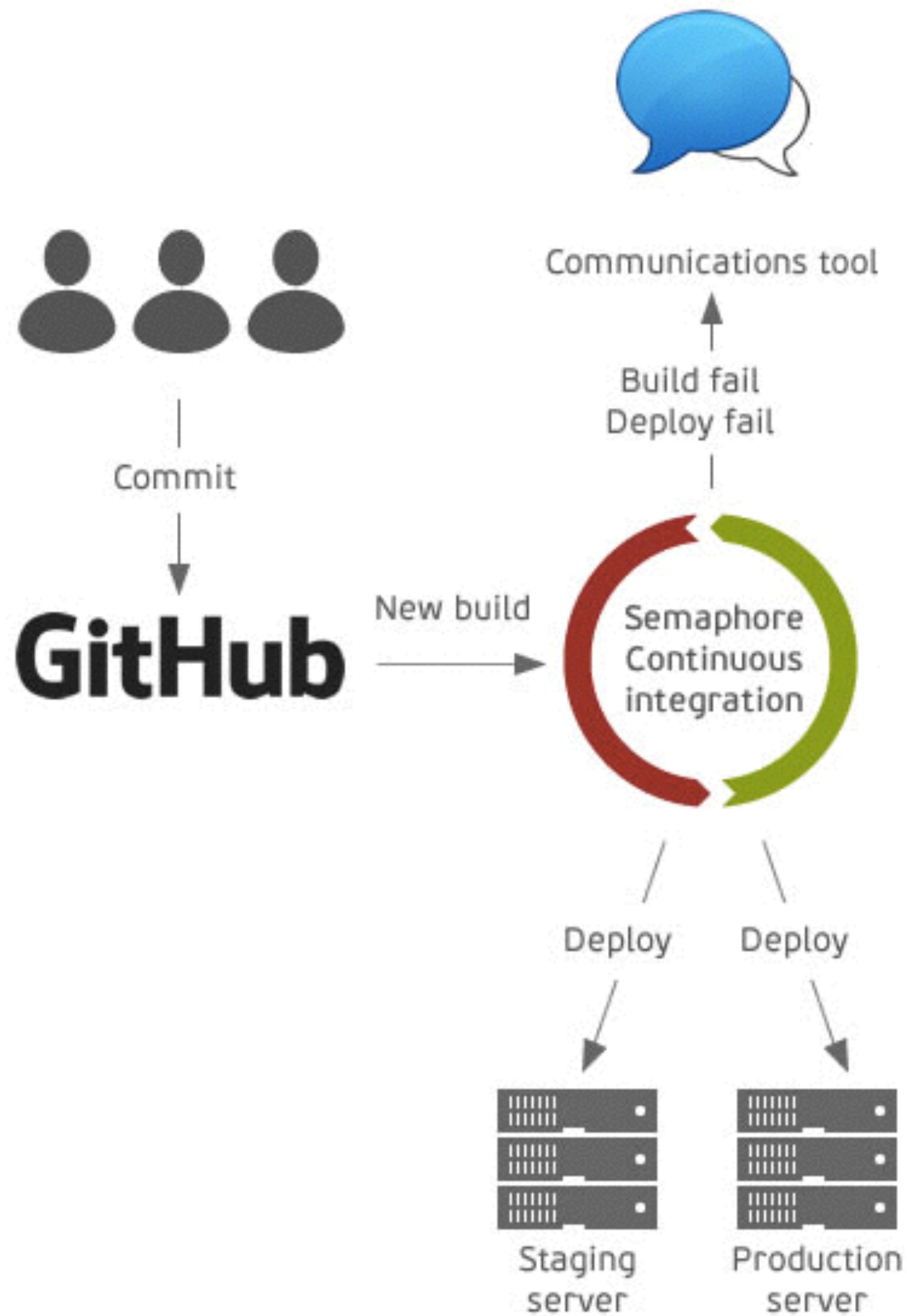
```
$ git clone git@github.com:geraldvillorente/drupal-pre-commit.git  
$ cp drupal-pre-commit/pre-commit /path/to/my/project/.git/  
$ sudo chmod +x /path/to/my/project/.git/pre-commit
```

HOW DO I FIT ALL OF THIS
INTO MY PROJECT?

Continuous Integration (CI) !

WHAT IS CI?

- The frequent integration of new code into a shared repository, where automated builds are performed to detect problems early.
- CI tools generally wait for some trigger, and then execute a set of scripts.
- Popular tools (available as SaaS):
 - Jenkins
 - TravisCI



EXAMPLE DRUPAL BUILD

1. Check GitHub for new commits every 10 minutes. Or setup webhooks.

2. Push new code to integration server.

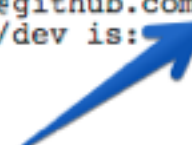
to-ace > GitHub Hook Log

[ENABLE AUTO REFRESH](#)

Last GitHub Push

```
Started on Mar 16, 2015 10:27:51 AM
Using strategy: Default
[poll] Last Built Revision: Revision e5e42bfa3c47772ce6792dc3604506e006a5ecf9 (github/dev, acquia/dev)
using GIT_SSH to set credentials Private key
> git --version # timeout=10
> git -c core.askpass=true ls-remote -h [redacted]@svn-11692.prod.hosting.acquia.com:[redacted].git # timeout=10
using GIT_SSH to set credentials Private key
> git -c core.askpass=true ls-remote -h git@github.com:[redacted] # timeout=10
[poll] Latest remote head revision on github/dev is: e42bfa3c47772ce6792dc3604506e006a5ecf9 - already built by 146
Done. Took 0.58 sec
No changes
```

A push to GitHub will trigger a push to Acquia Cloud



3. Run updb, clear caches, revert features on integration server.

update-db:

```
[echo] drush @.dev updb
[exec] Current OS is Linux
[exec] Executing '/scratch/jenkins/workspace/dev-smoke/vendor/bin/drush' with arguments:
[exec] '@.dev'
[exec] 'updb'
[exec] '-y'
[exec]
[exec] The ' characters around the executable and arguments are
[exec] not part of the command.
[exec] No database updates required [success]
[exec] 'all' cache was cleared. [success]
[exec] Finished performing updates. [ok]
```

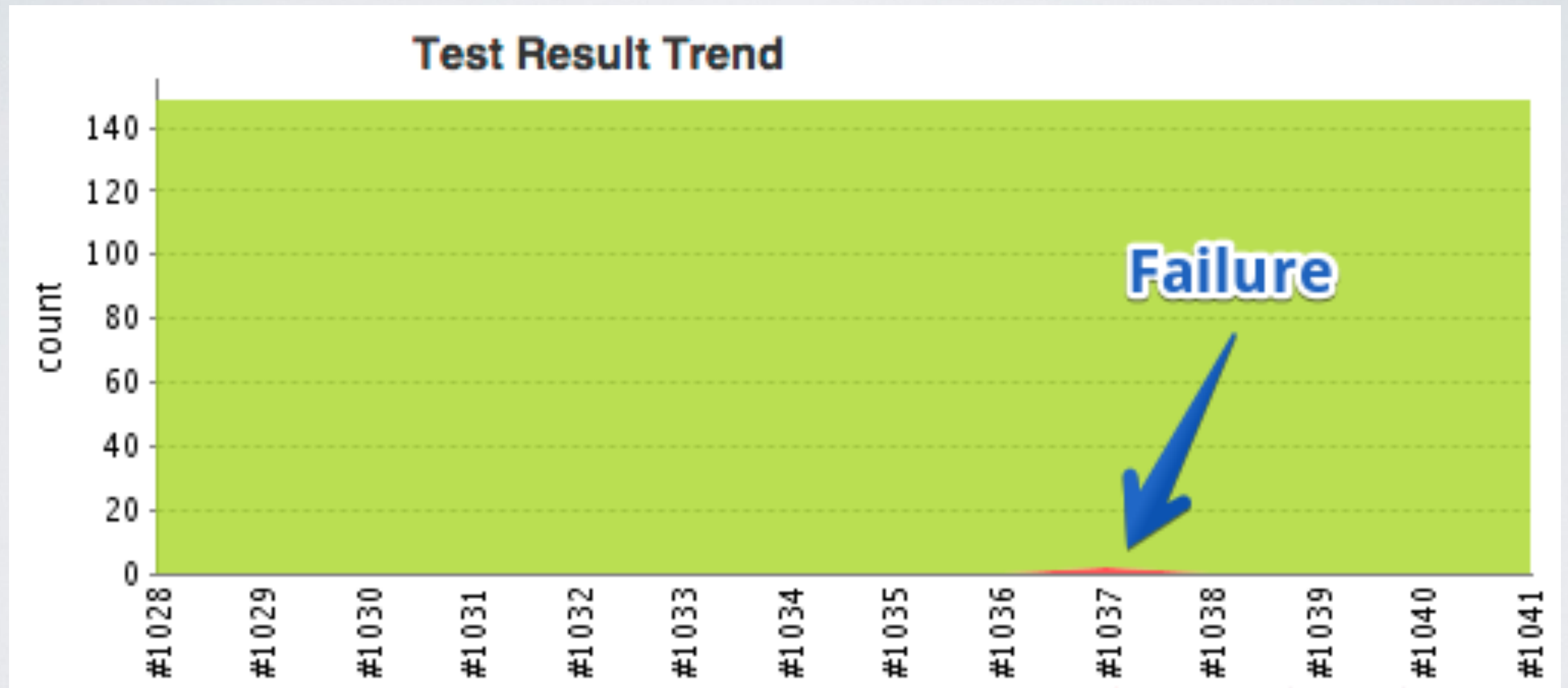
clear-caches:

```
[echo] drush @atf.dev cc
[exec] Current OS is Linux
[exec] Executing '/scratch/jenkins/workspace/dev-smoke/vendor/bin/drush' with arguments:
[exec] '@atf.dev'
[exec] 'cc'
[exec] 'all'
[exec] '-y'
[exec]
[exec] The ' characters around the executable and arguments are
[exec] not part of the command.
[exec] 'all' cache was cleared. [success]
```

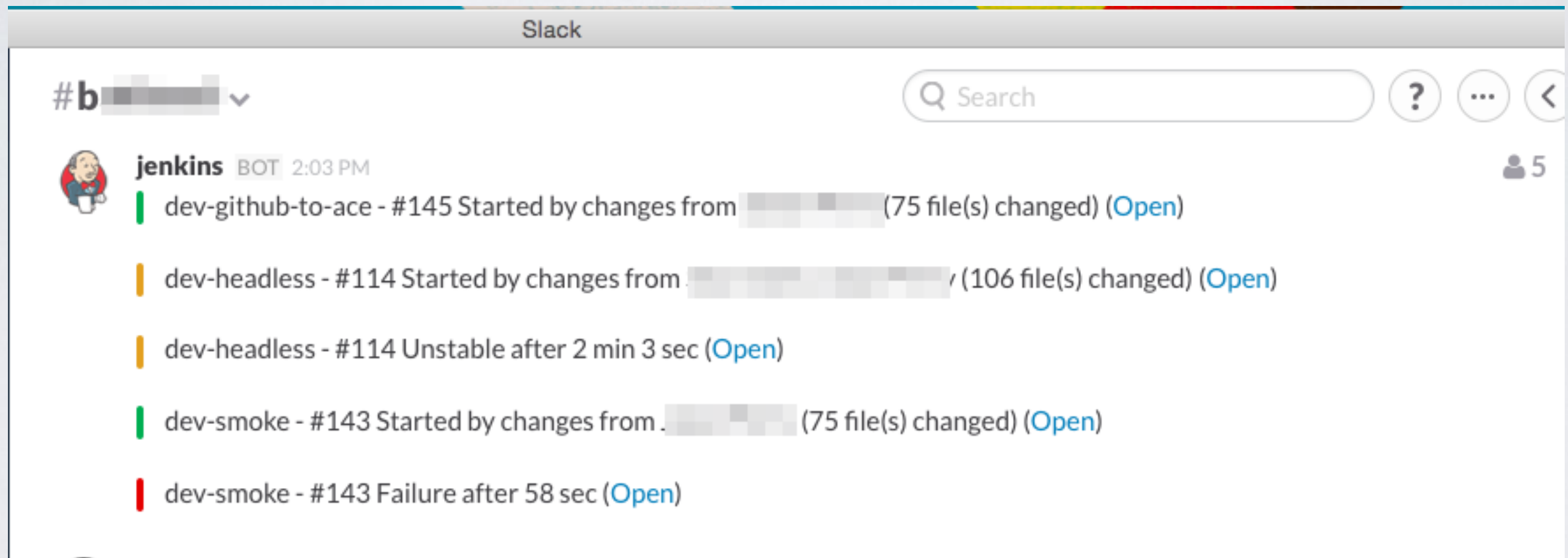
features-revert:

```
[echo] drush @.dev fra
[exec] Current OS is Linux
[exec] Executing '/scratch/jenkins/workspace/dev-smoke/vendor/bin/drush' with arguments:
[exec] '@.dev'
[exec] 'fra'
```

4. Run Behat tests on integration server.



5. Post message into chat room reporting SUCCESS/FAILURE



The screenshot shows a Slack chat window titled "Slack". The channel name is "#b" followed by a blurred name and a dropdown arrow. A search bar with a magnifying glass icon and the word "Search" is on the right. Below the channel header, the Jenkins bot (jenkins BOT) has sent a message at 2:03 PM. The message contains five lines of build status reports, each preceded by a colored vertical bar: green for success, orange for unstable, and red for failure. Each line includes the build name, number, start reason, duration, and a link to view details.

#b [blurred name] ▾

Q Search ? ... <

jenkins BOT 2:03 PM 5












- dev-github-to-ace - #145 Started by changes from [blurred name] (75 file(s) changed) ([Open](#))
- dev-headless - #114 Started by changes from [blurred name] (106 file(s) changed) ([Open](#))
- dev-headless - #114 Unstable after 2 min 3 sec ([Open](#))
- dev-smoke - #143 Started by changes from [blurred name] (75 file(s) changed) ([Open](#))
- dev-smoke - #143 Failure after 58 sec ([Open](#))

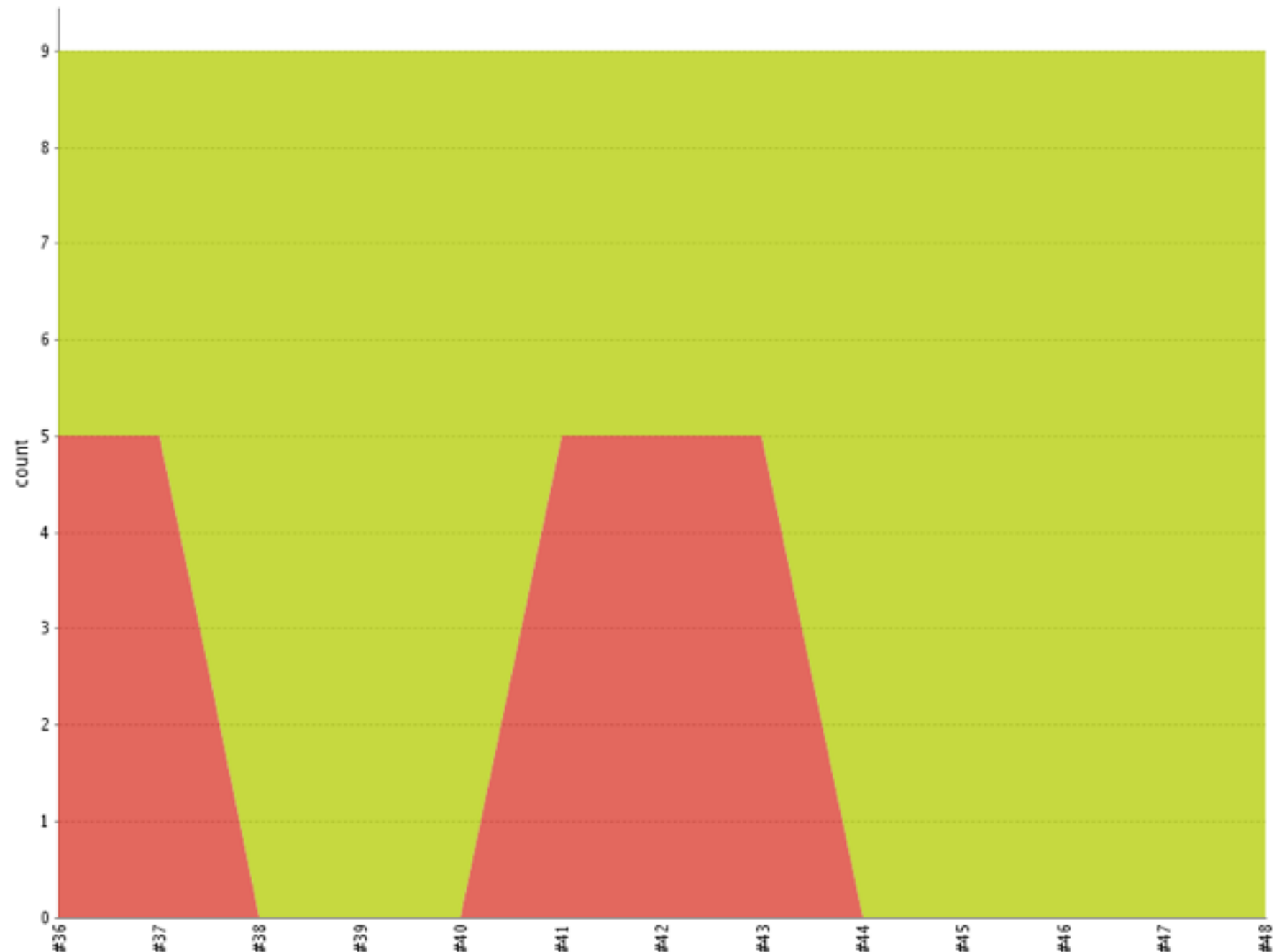
EXAMPLE BUILD REPORT

Jenkins ▾ > smoke-debug > Test Report

[ENABLE AUTO REFRESH](#)

- [Back to Dashboard](#)
- [Status](#)
- [Changes](#)
- [Workspace](#)
- [Build Now](#)
- [Delete Project](#)
- [Configure](#)
- [GitHub](#)
- [Groups](#)
- [Roles](#)
- [Move](#)

Build History			trend ▬
	#48	Mar 7, 2015 7:59:02 PM	30 KB
	#47	Mar 7, 2015 7:41:31 PM	30 KB
	#46	Mar 7, 2015 6:54:58 PM	30 KB
	#45	Mar 7, 2015 6:48:53 PM	39 KB
	#44	Mar 7, 2015 6:46:10 PM	39 KB
	#43	Mar 7, 2015 6:43:08 PM	57 KB
	#42	Mar 7, 2015 6:33:22 PM	47 KB
	#41	Mar 7, 2015 6:31:01 PM	47 KB
	#40	Mar 7, 2015 6:26:51 PM	30 KB
	#39	Mar 7, 2015 6:22:59 PM	30 KB
	#38	Mar 7, 2015 6:19:43 PM	30 KB



EXAMPLE TEST FAILURES

Test Result

5 failures (±0)

9 tests (±0)

Took 26 sec.

 [add description](#)

All Failed Tests

Test Name	Duration	Age
+ Careers.Legal Careers page displays vacancy announcements and search filters	21 sec	2
+ Home Page.Slideshow is visible	0.3 sec	2
+ Home Page.Action links is pane visible	0.21 sec	2
+ Follow Us.Follow links are vsisible	0.2 sec	2
+ USASearch module.Front Page search block returns results	0.15 sec	2

All Tests


Package	Duration	Fail	(diff)	Skip	(diff)	Pass	(diff)	Total	(diff)
(root)	26 sec	5		0		4		9	

EXAMPLE TEST FAILURES

- Worried about quality? Fail a build.



Checkstyle: [4 warnings](#) from one analysis.




- [4 new warnings](#)
- Plug-in Result:  - [4 warnings](#) exceed the threshold of 0 by 4 (Reference build: [#142](#))

Checkstyle Warnings - New Warnings

Summary

Total	High Priority	Normal Priority
4	3	1

Details

Categories	Types	Warnings	Details	High	Normal
Category	Total		Distribution		
FileEnd	1				
LineLength	1				
SuperfluousWhitespace	2				
Total	4				

CI RESOURCES

- Jenkins - <http://jenkins-ci.org>
- Travis - <https://travis-ci.org>
- Bamboo - <https://www.atlassian.com/software/bamboo>
- <https://github.com/unnn/test-driven-drupal-qa>

GET STARTED TODAY

- That last link on that last side...
- <https://github.com/unnn/test-driven-drupal-qa>

branch: master ▾

test-driven-drupal-qa / build.xml



unn on Aug 12, 2014 WIP

1 contributor

449 lines (388 sloc) | 16.506 kb

Raw

Blame

History



```
1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <project name="Drupal" default="build-full">
4   <property environment="env"/>
5   <property name="sites-dir" value="${basedir}/sites/all/modules"/>
6   <property name="bin-dir" value="${basedir}/vendor/bin"/>
7   <property name="log-dir" value="${basedir}/build/logs"/>
8   <property file="${basedir}/build.properties"/>
9   <taskdef resource="net/sf/antcontrib/antlib.xml">
10     <classpath>
11       <pathelement location="${basedir}/scripts/ant-contrib-1.0b3.jar"/>
12     </classpath>
13   </taskdef>
14
```

THE GLUE

Invoke Ant

Targets

Advanced...

Delete

Connect it up with Jenkins

Add build step

TIPS

- Integrate with PM tool. E.g., JIRA
- Utilize webhooks on SCM. E.g., GitHub
- Make builds repeatable (both locally and in CI)
- Fail fast

Thank you!

Questions?

“If you don’t like testing your product, most likely your customers won’t like to test it either.”

-Anonymous