

# Continuous Deployment

**Sam Haskins**

**Etsy.com**

**[shaskins@etsy.com](mailto:shaskins@etsy.com)**

**Velocity China 2012**

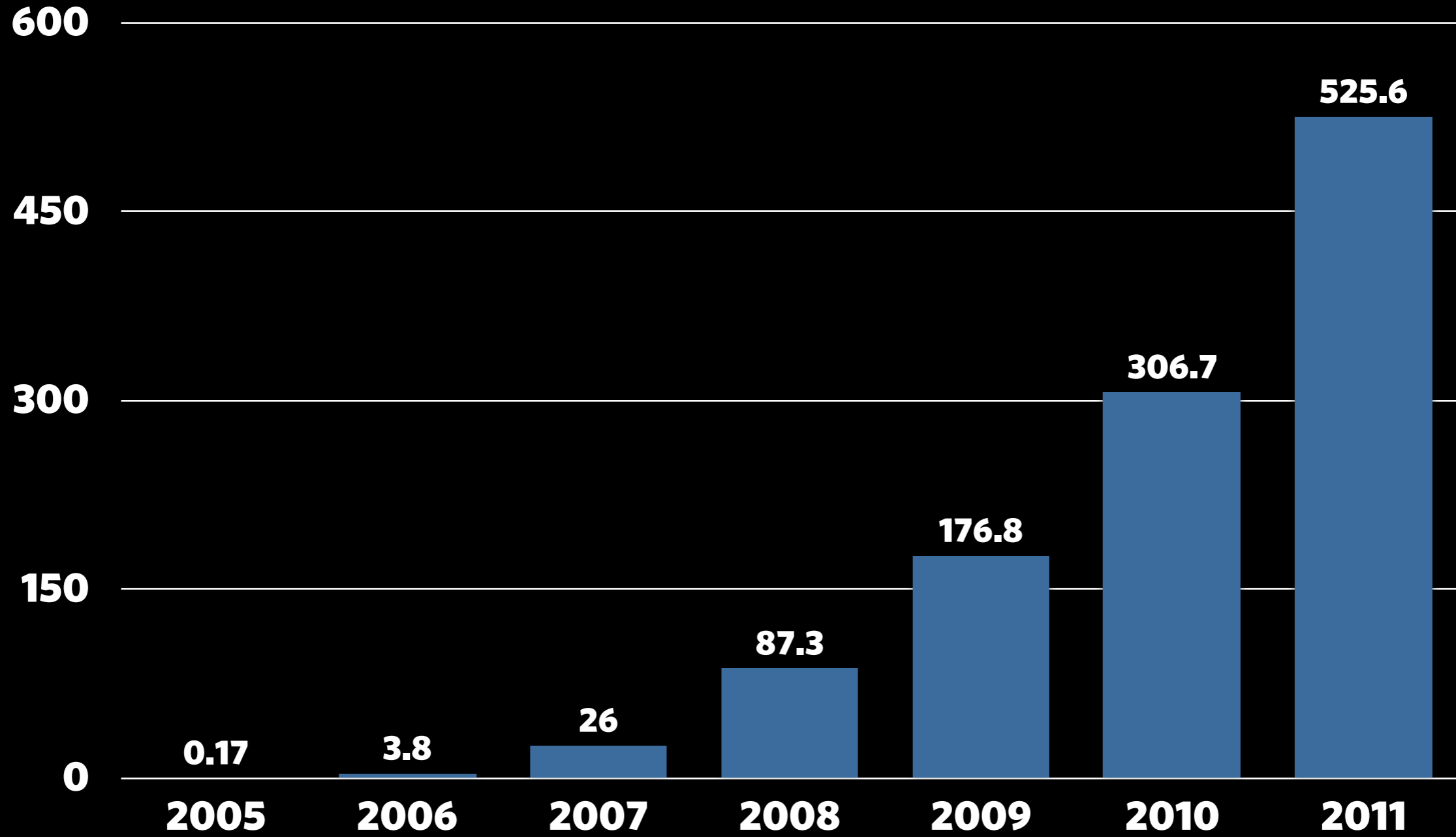
Etsy

**20,000,000 Members**

**800,000 Active Shops**

**17,000,000 items**

■ Millions of \$



■ Millions of \$

600

450

300

150

0

2005

2006

2007

2008

2009

2010

2011

0.17

3.8

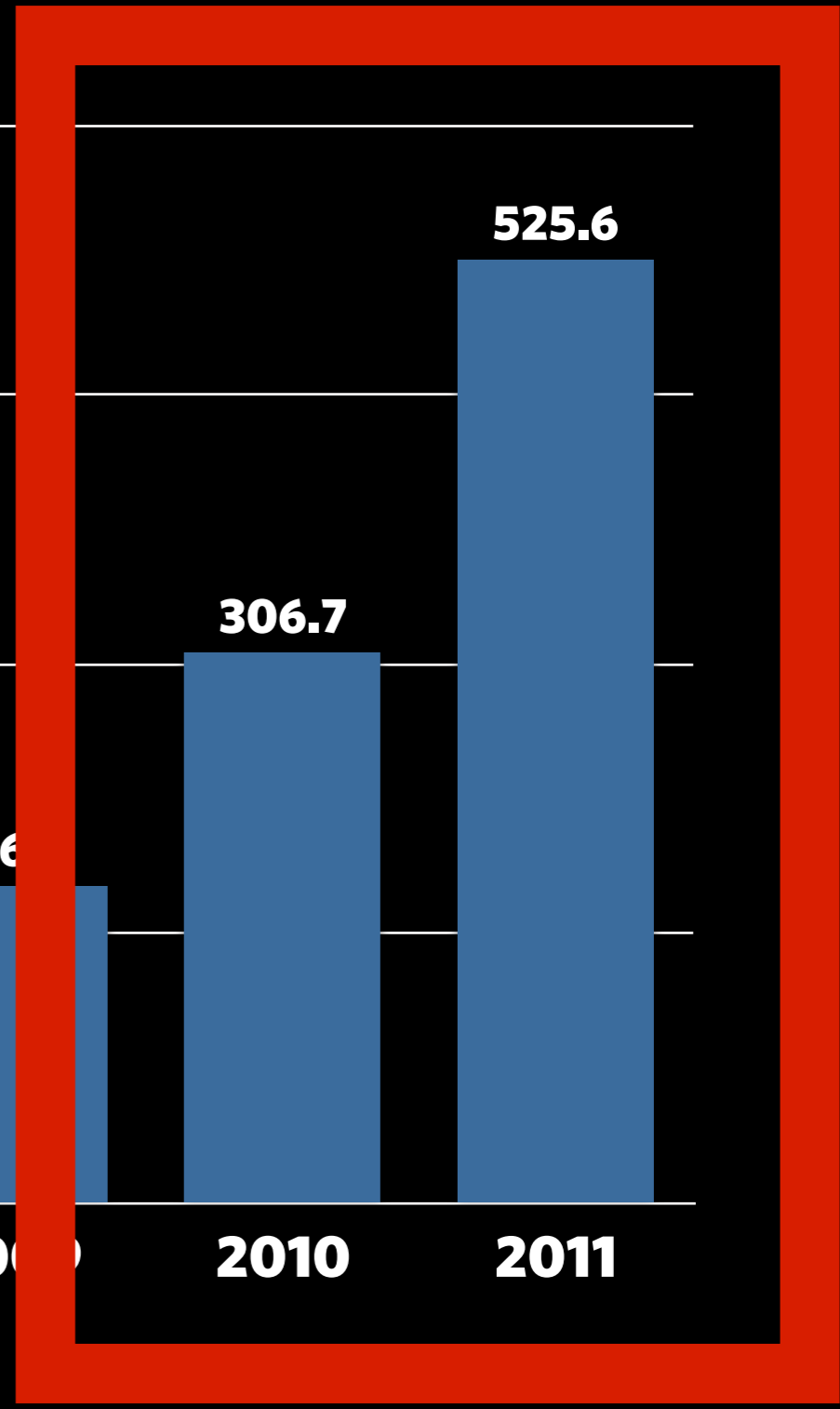
26

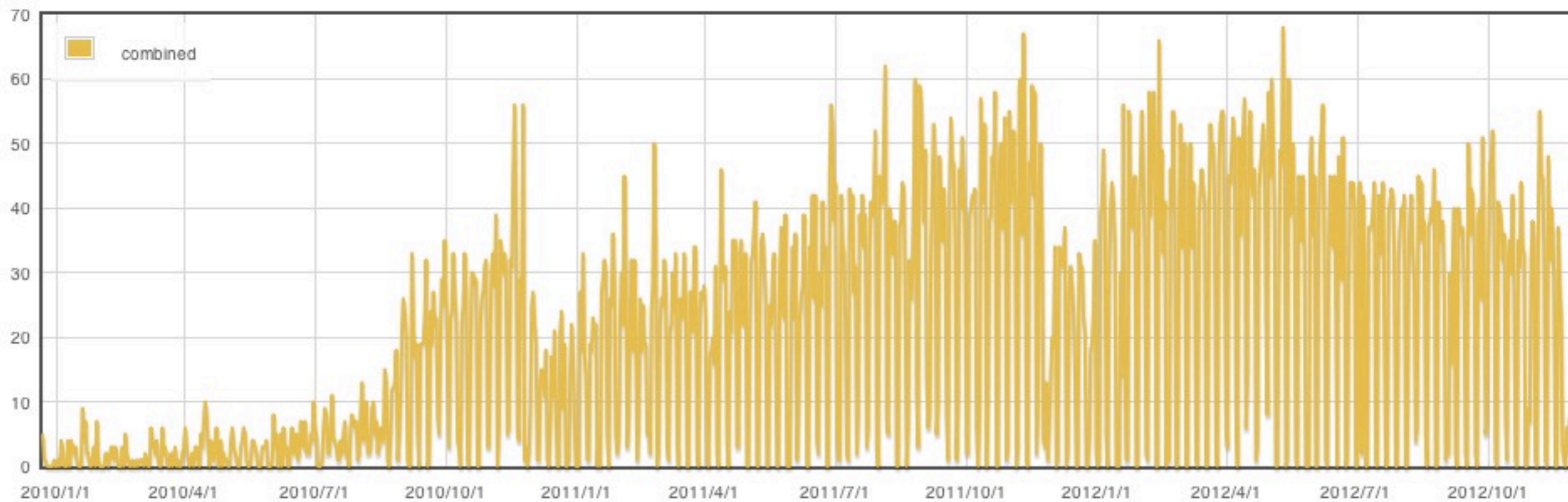
87.3

176

306.7

525.6





**(deploys/day)**



■ Millions of \$

600

450

300

150

0

2005

2006

2007

2008

2009

2010

2011

0.17

3.8

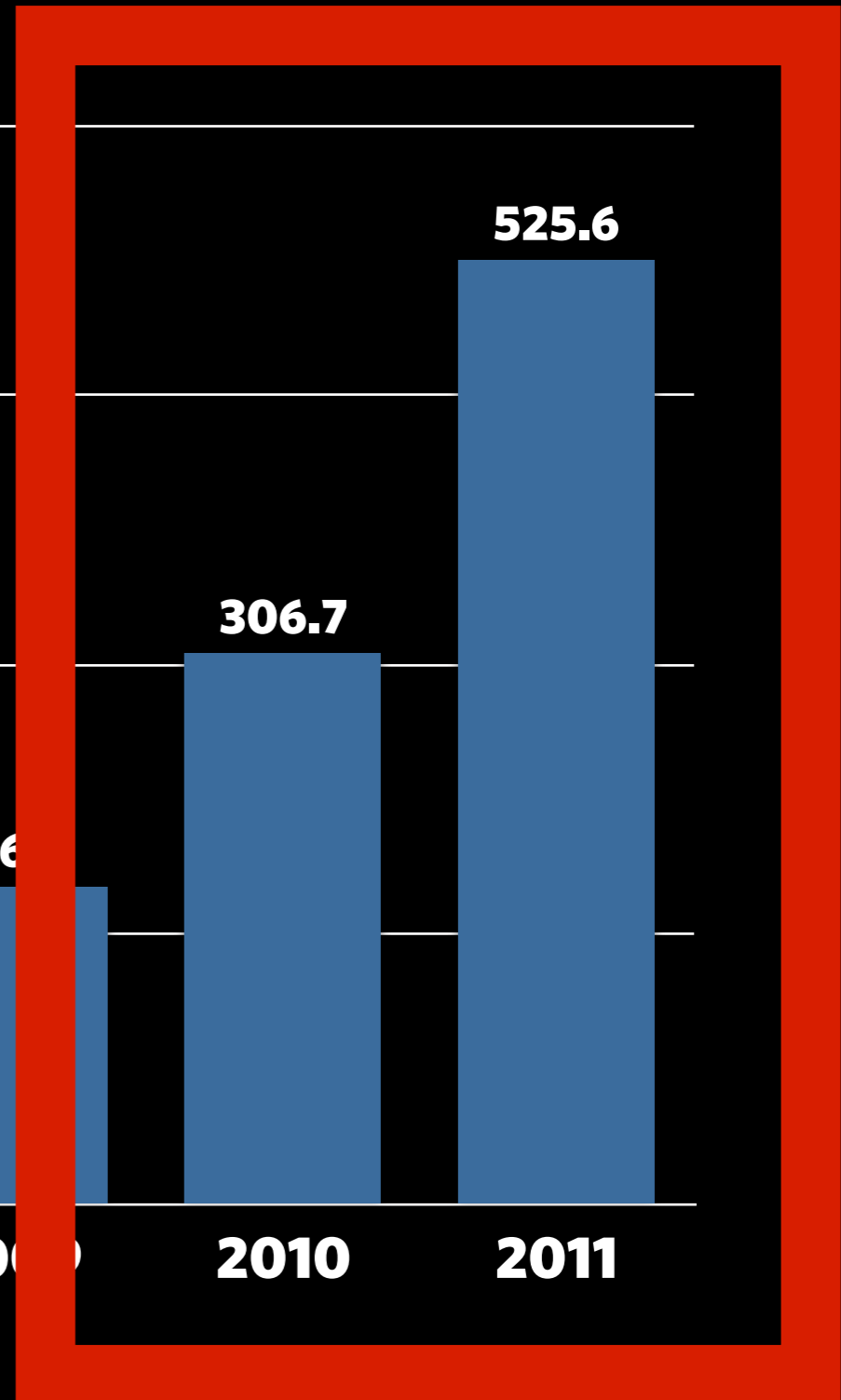
26

87.3

176

306.7

525.6



**200 deployers**

**200** **deployers**

**Engineers?**

**200** **deployers**

**Engineers +  
Designers**

# 200 ~~deployers~~

**Engineers +  
Designers +  
Product**

**33 deploys/day**

**33 deploys/day**

**(average that includes weekends)**

**33 deploys/day**

**(average that includes weekends)**

**(just the web stack)**



# Continuous Deployment

**Take the ceremony  
out of releases**

**Smaller changes,  
more often**

**Fewer changes/deploy**

**=**

**Fewer things that  
can go wrong**

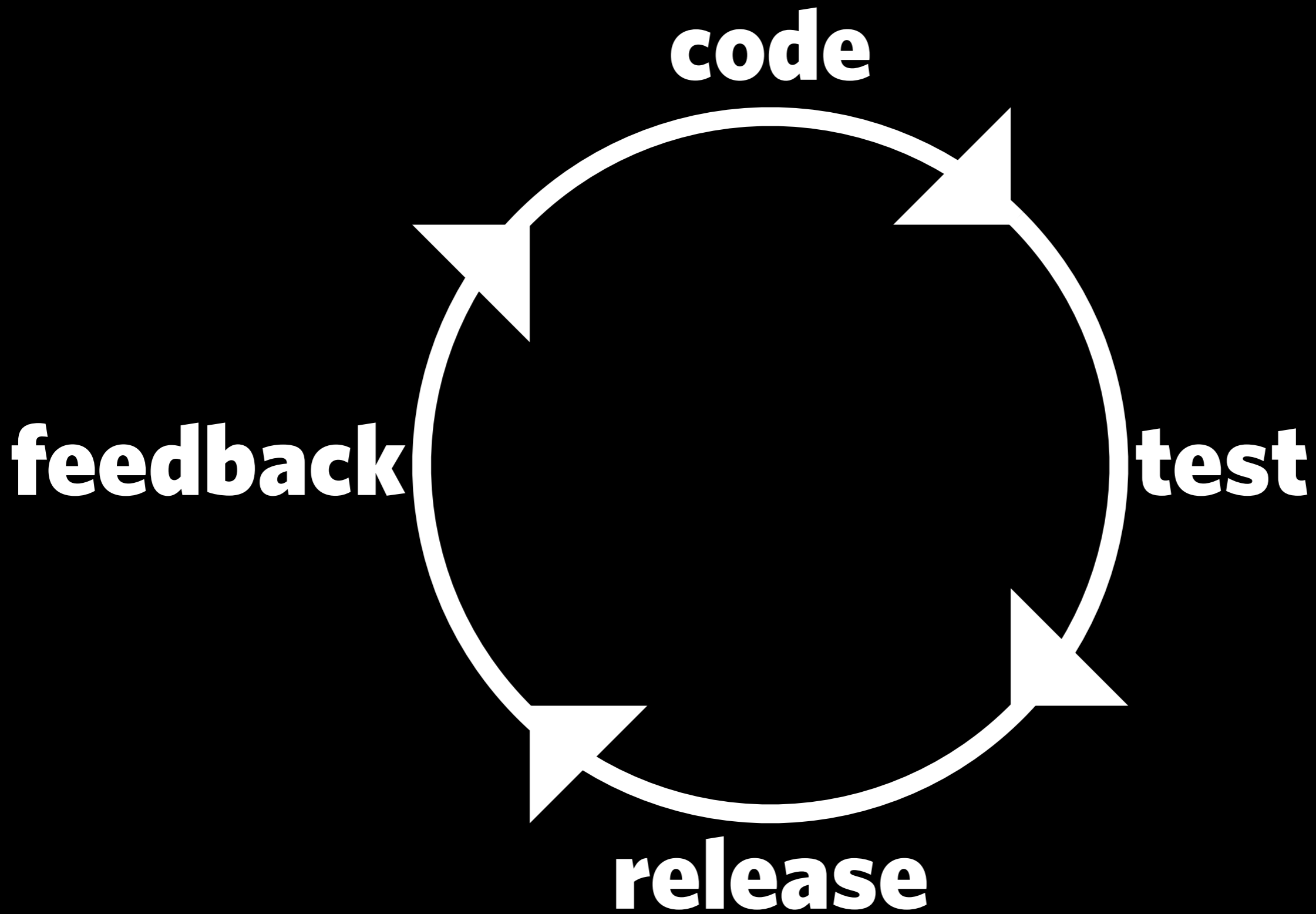
**Fewer changes/deploy**

**=**

**Easier to see effects of  
your change**

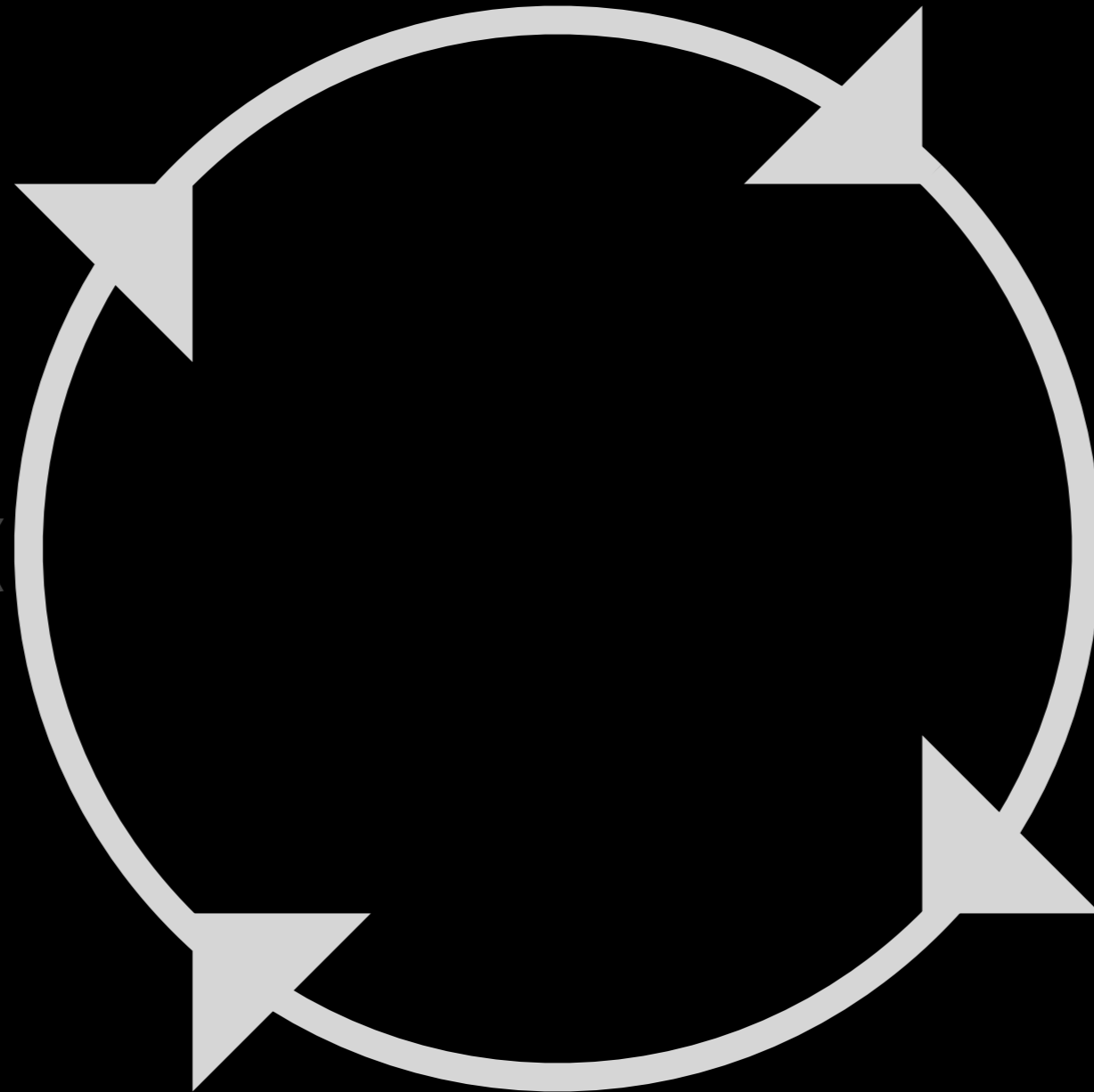


# How to implement





**code**



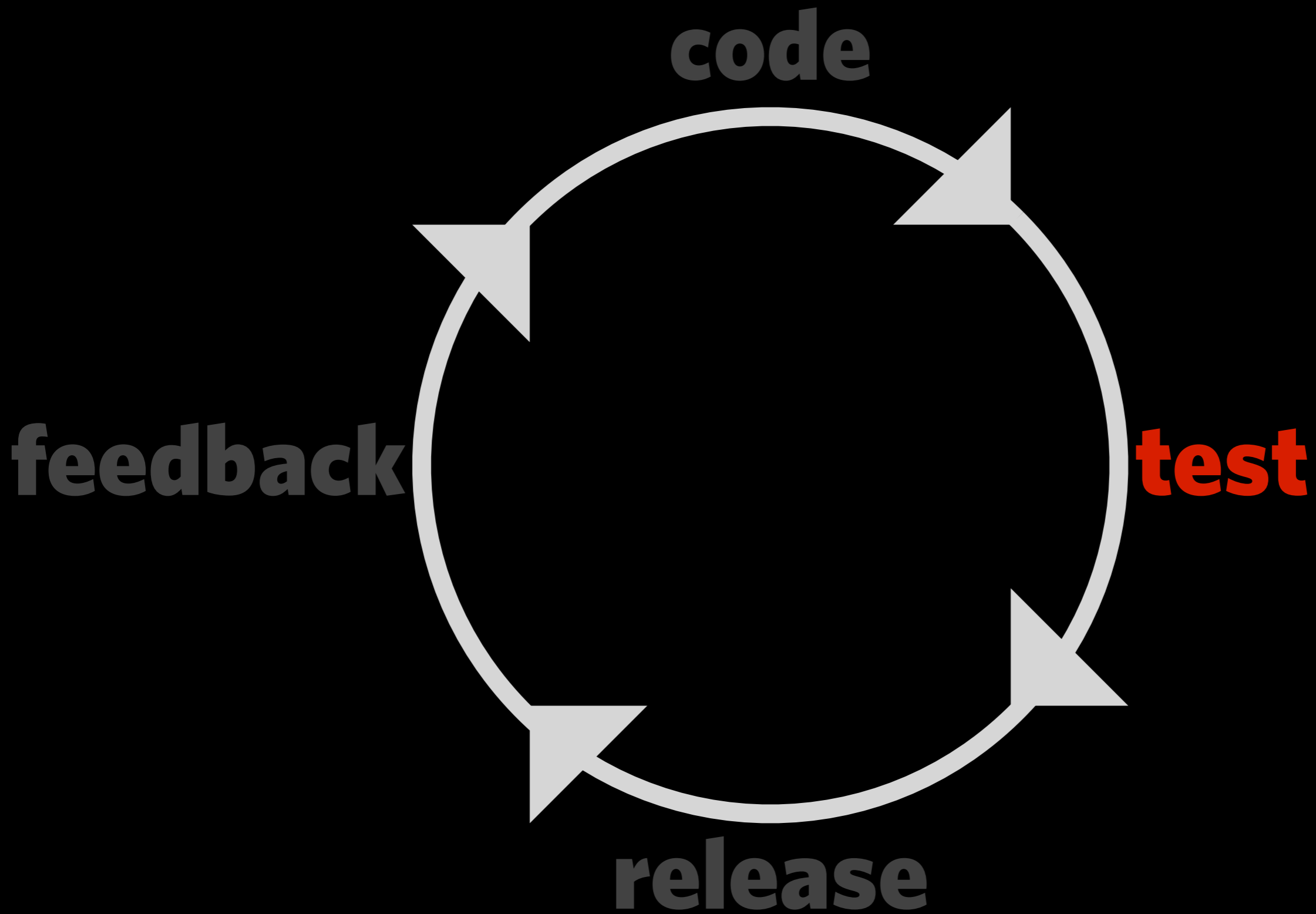
**feedback**

**test**

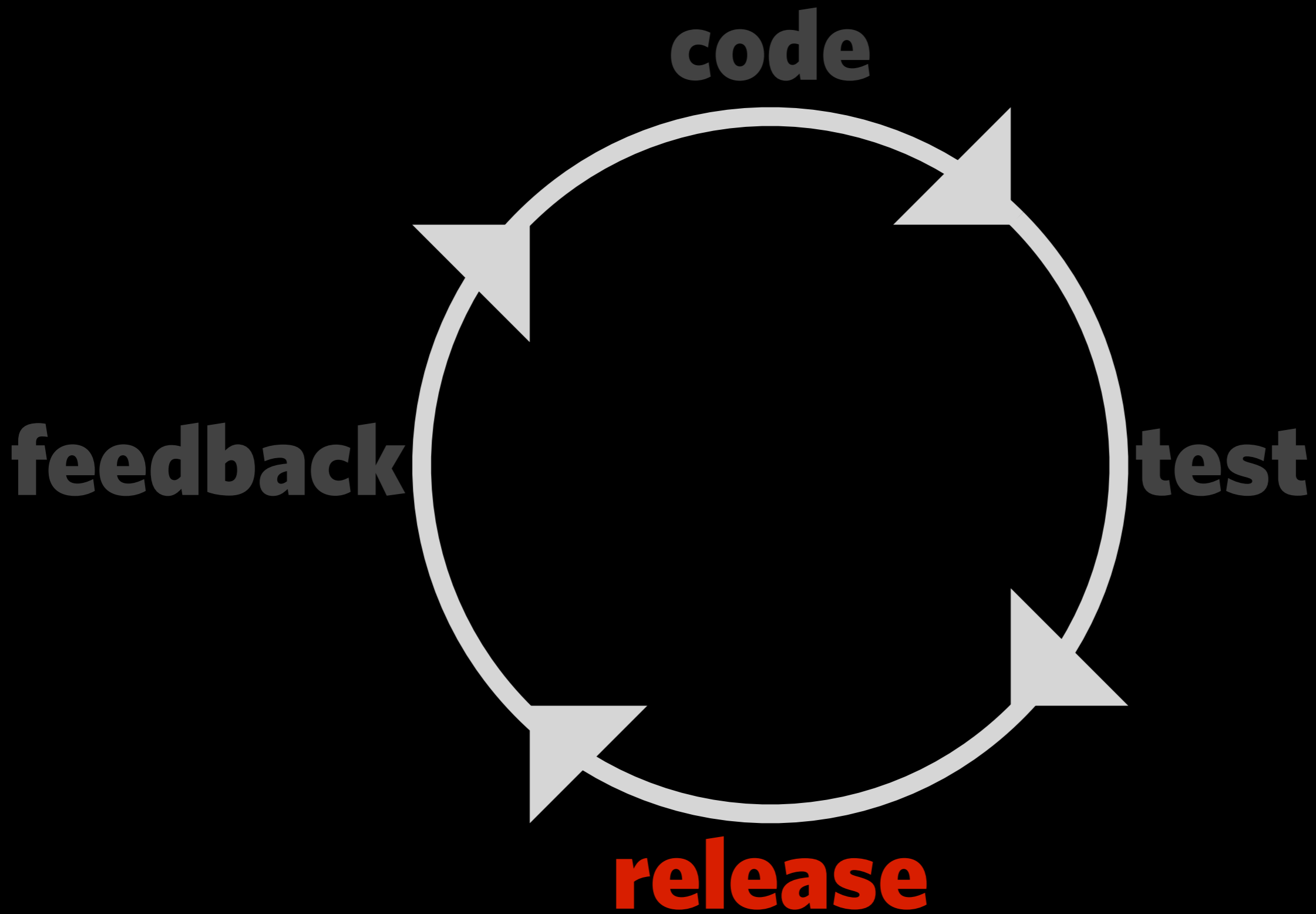
**release**

```
<?php
// search engine

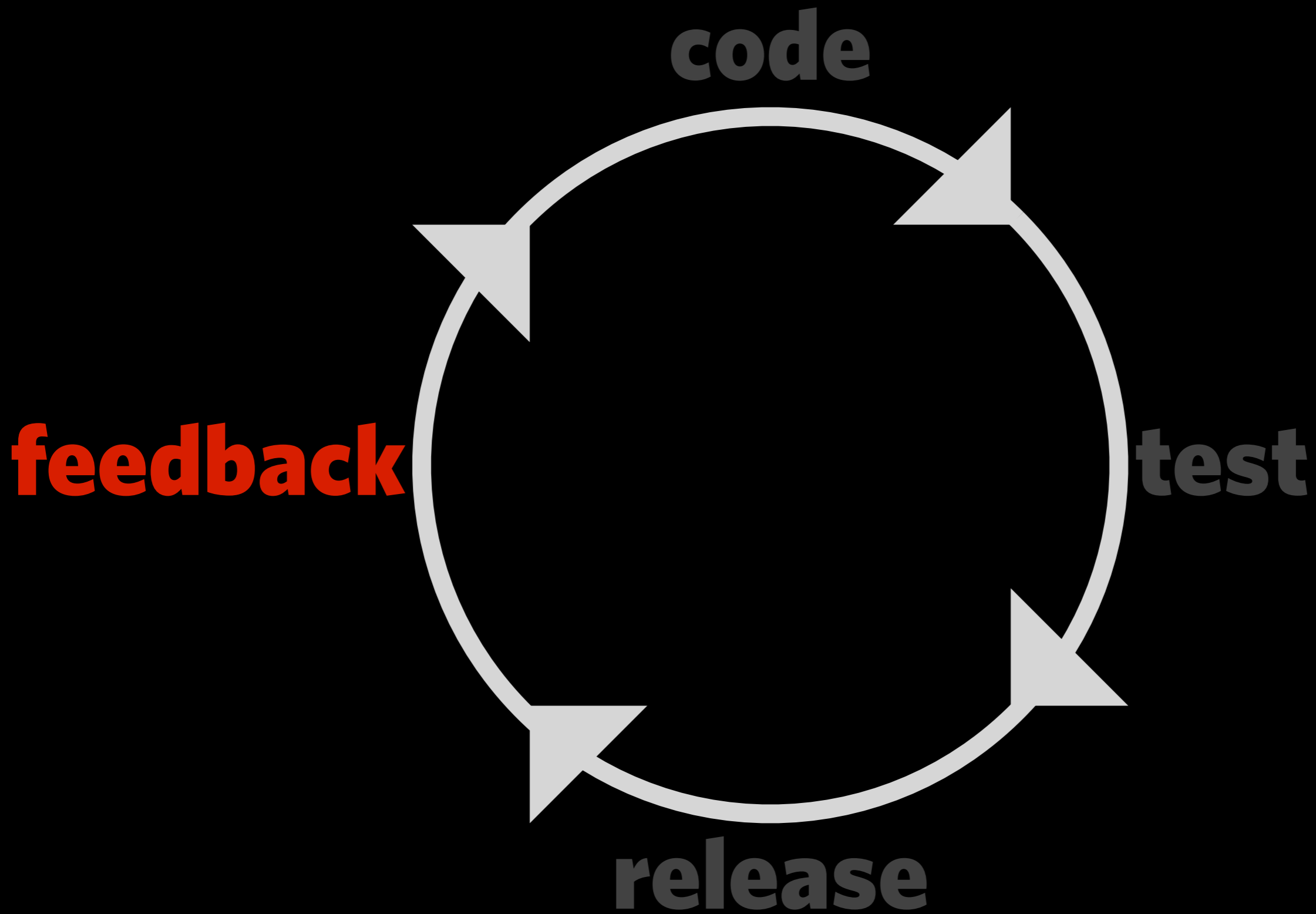
$results = Search::use_grep($_GET['query']);
print_r($results);
```



```
$ curl http://shaskins.dev.etsy.com/search.php?query=bowties
```



```
$ scp -r ~/code webserver.prod.etsy.com:/docroot
```

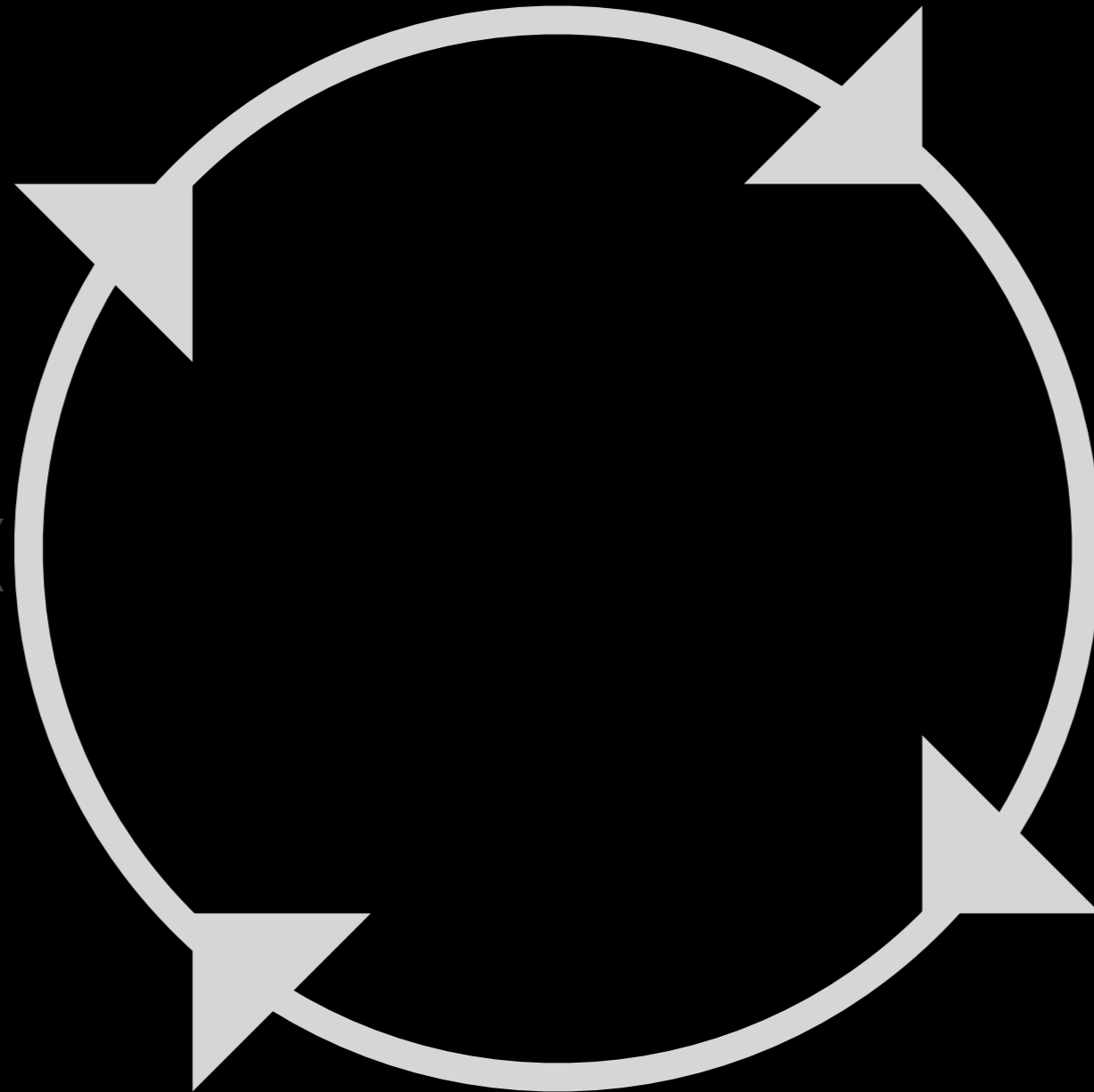


```
$ tail -f access_log
[15/Nov/2012:17:14:41 +0000] "GET /search?query=bowties" 200
[15/Nov/2012:17:14:42 +0000] "GET /search?query=knit+hat" 200
[15/Nov/2012:17:14:45 +0000] "GET /search?query=vintage+chair"
```





**code**



**feedback**

**test**

**release**

# **"Always Ship Trunk"**

**Paul Hammond**

<http://www.paulhammond.org/2010/06/trunk/>

# **Centralized configuration file**

```
<?php
// Site configuration

$config['search'] = array(
    'enabled' => 'on'
);
```

```
<?php
// search engine
```

```
$results = Search::use_grep($_GET['query']);
print_r($results);
```

```
<?php
// search engine
if (!Feature::isEnabled('search')) {
    header('HTTP/1.1 503 Service Unavailable');
}

$results = Search::use_grep($_GET['query']);
print_r($results);
```

```
<?php
// Site configuration

$config['search_method'] = array(
    'grep' => '100%'

);
```



```
<?php
// Site configuration

$config['search_method'] = array(
    'grep' => '100%',
    'solr_service' => '0%'
);
```

```
<?php
// search engine
if (!Feature::isEnabled('search')) {
    header('HTTP/1.1 503 Service Unavailable');
    exit();
}

$results = Search::use_grep($_GET['query']);

print_r($results);
```

```
<?php
// search engine
if (!Feature::isEnabled('search')) {
    header('HTTP/1.1 503 Service Unavailable');
    exit();
}

switch (Feature::variant('search_method')) {
    case 'grep':
        $results = Search::use_grep($_GET['query']);
        break;

    case 'solr_service':
        $results = Search::use_solr($_GET['query']);
        break;
}

print_r($results);
```

**Lots of types of options**

```
<?php
// Site configuration

$config['new_feature'] = array(
    'enabled' => '0%',
    'admin' => 'on'
);
```

```
<?php
// Site configuration

$config['new_feature'] = array(
    'enabled' => '0%',
    'users' => array(
        'test_user',
        'big_seller',
        'best_friend'
    )
)
);
```

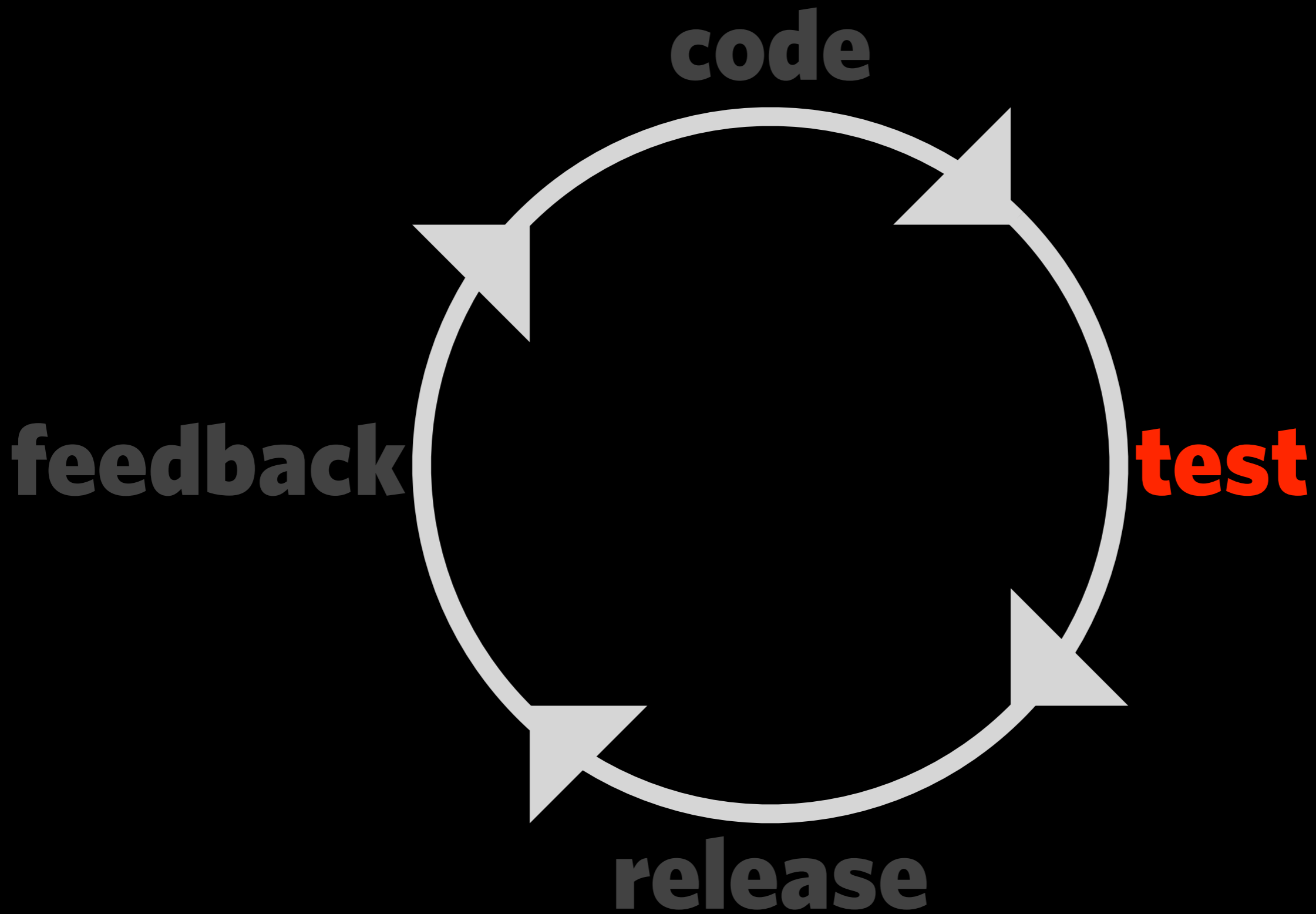
```
<?php
// Site configuration









$config['new_feature'] = array(
    'enabled' => '0%',
    'groups' => array(
        1234,
        6742
    )
);
```

```
<?php
// Site configuration


$config['new_feature'] = array(
    'enabled' => array(
        'variant_1' => '15%',
        'variant_2' => '20%',
        'control'   => '65%'
    )
)
);
```





-  [Back to Project](#)
-  [Status](#)
-  [Changes](#)
-  [Console Output](#)
-  [View Build Information](#)
-  [Parameters](#)
-  [Previous Build](#)
-  [Next Build](#)

## Build #38125 (Nov 30, 2012 7:36:41 PM)

Progress: 

### shaskins



No changes.



Started by command line for [shaskins](#)

### Latest Builds

	Build	Date
	<a href="#">try-hphp #30973</a>	3 min 41 sec
	<a href="#">try-orm-tests #30858</a>	3 min 41 sec
	<a href="#">try-integration-tests #30772</a>	3 min 41 sec
	<a href="#">try-validate-css #35085</a>	3 min 41 sec
	<a href="#">try-unit-tests #35845</a>	3 min 41 sec
	<a href="#">try-build-js #35519</a>	3 min 41 sec
	<a href="#">try-php-code-sniffer #35711</a>	3 min 41 sec
	<a href="#">try-phpunit-code-sniffer #24462</a>	3 min 41 sec
	<a href="#">try-replication-tests #32367</a>	3 min 41 sec
	<a href="#">try-build-templates #35271</a>	3 min 41 sec
	<a href="#">try-gci-tests #35151</a>	3 min 41 sec



**Development**

**Staging**

**Production**

# Development

# Development

**Separate database**

# Development

**Separate database**

**Separate network**

# Staging and Production

# **Staging and Production**

**Same databases**



# Staging and Production

**Same databases**

**Same caches**

# Staging and Production

**Same databases**

**Same caches**

**Same services**

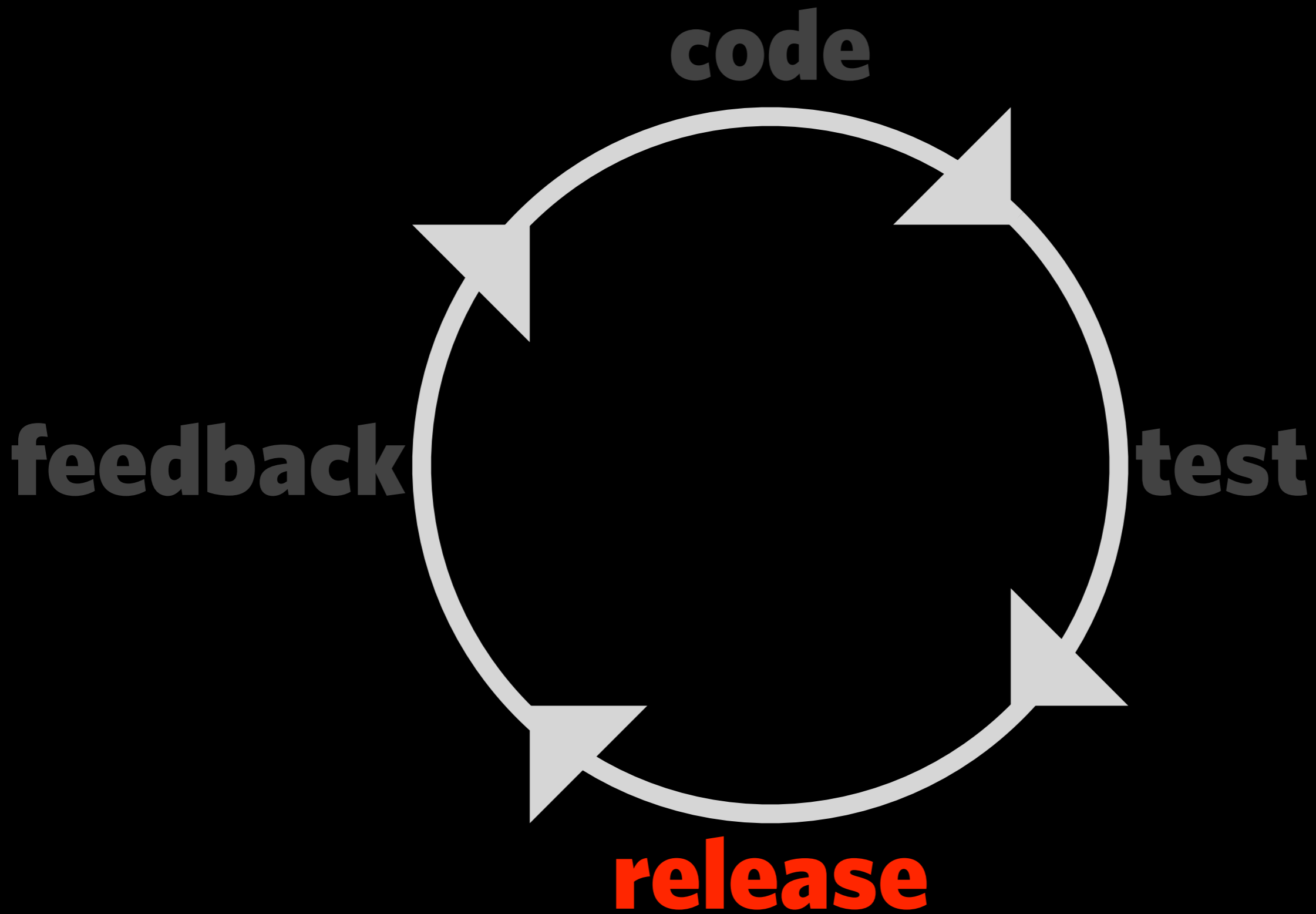
# Staging and Production

**Same databases**

**Same caches**

**Same services**

**Same hardware**





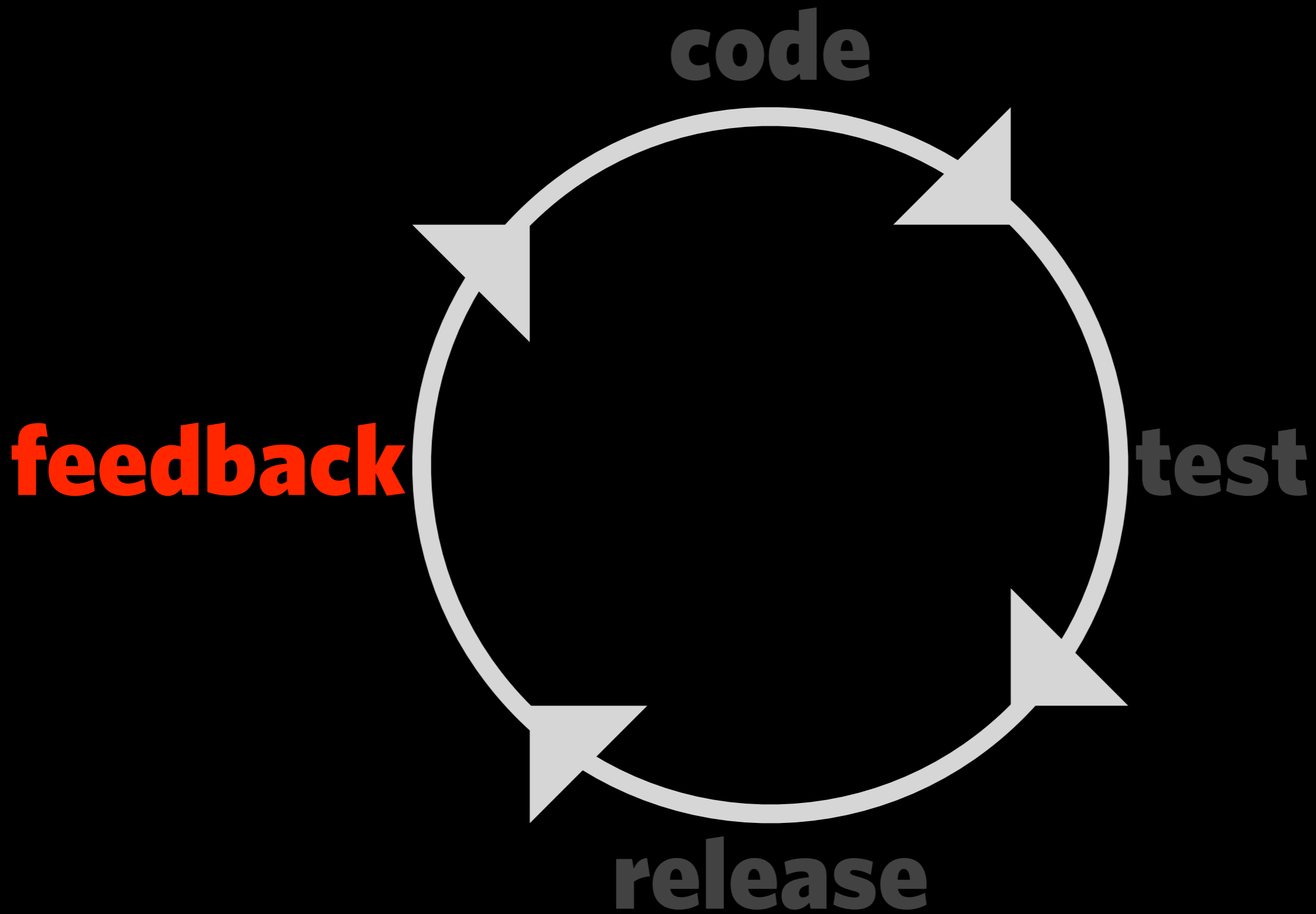
## 02. Deploy to Production

0cc91de → 0cc91de

**Deploy to Production**

What to watch after a push  
Log Watcher (supergrep)  
Forums

**Configuration-only  
deploys**





# Metrics

# Metrics

**Hundreds of  
thousands of metrics**

# **Business Metrics**

# **Business Metrics**

**Number of sales**

# **Business Metrics**

**Number of sales**

**Dollars of revenue**

# **Business Metrics**

**Number of sales**

**Dollars of revenue**

**Sources of traffic**

# Application Metrics

# **Application Metrics**

**Application actions**



# **Application Metrics**

**Application actions**

**Usage patterns**

# **Application Metrics**

**Application actions**

**Usage patterns**

**Times events happen**

# System Metrics

# **System Metrics**

**CPU usage**

# **System Metrics**

**CPU usage**

**Network bandwidth**

# **System Metrics**

**CPU usage**

**Network bandwidth**

**Disk IO**

**Make metrics  
easy to create**

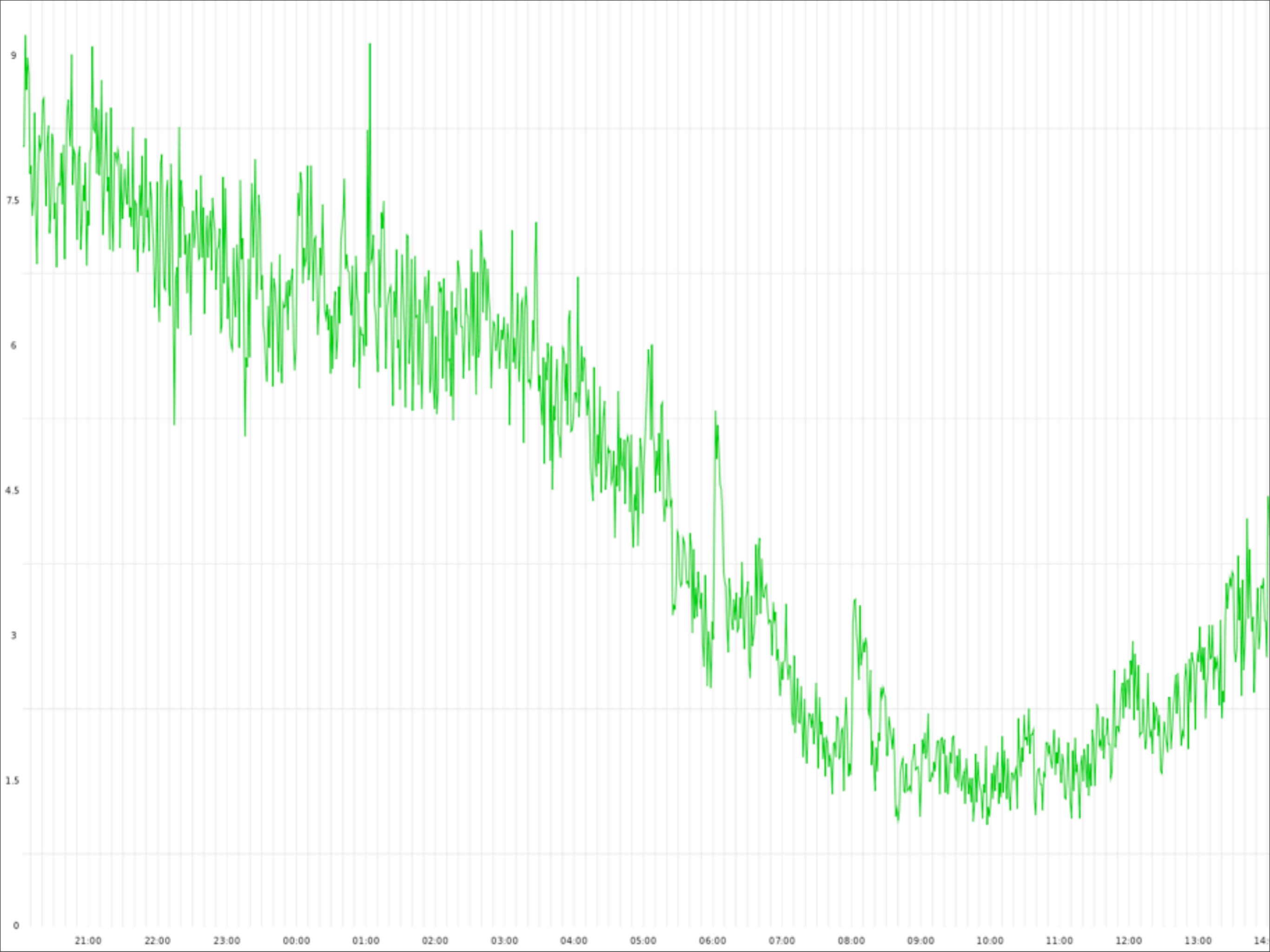
```
<?php
// search engine

$results = Search::use_grep($_GET['query']);
print_r($results);
```



```
<?php
// search engine

$results = Search::use_grep($_GET['query']);
print_r($results);
StatsD::increment('search.queries');
```



Friday, December 21, 12

**Put them into  
dashboards**

# Logs



[Guide](#)[Legend](#)[Clear Log](#)[\[X\]](#)[Highlight](#)[\[?\]](#)

# Supergrep++

Lost connection... retrying in 64 seconds

Lost connection... retrying in 32 seconds

Lost connection... retrying in 16 seconds

Lost connection... retrying in 4 seconds

Lost connection... retrying in 2 seconds

localhost [12/1 17:29:15] **error** [W0Bt8zHWbgNhP-jmnKYT0N-ZEHGA] [GENERAL] [Raw Log](#) PHP Fatal error: Allowed memory size of 67108864 bytes exhausted (tried to allocate 71 bytes) in /home

[01-Dec-2012 22:29:15 UTC] PHP Fatal error: Allowed memory size of 67108864 bytes exhausted (tried to allocate 71 bytes) in /home

localhost [12/1 22:29:15] **debug** [W0Bt8zHWbgNhP-jmnKYT0N-ZEHGA] [RAWSQL] [Raw Log](#) Success: 1, Interpolated Query: select `Taste

localhost [12/1 22:29:15] **debug** [W0Bt8zHWbgNhP-jmnKYT0N-ZEHGA] [ETSYORM\_DBCONNECTION] [Raw Log](#) Connecting to mysql:host=devmys

[Sat Dec 01 22:29:15 2012] [W0Bt8zHWbgNhP-jmnKYT0N-ZEHGA] [info] [Cache\_Memcached] [0] Memcache host: memcache.shaskins.vm.ny4dev

[Sat Dec 01 22:29:15 2012] [W0Bt8zHWbgNhP-jmnKYT0N-ZEHGA] [info] [Cache\_Memcached] [0] Memcache host: memcache.shaskins.vm.ny4dev

localhost [12/1 22:29:15] **debug** [W0Bt8zHWbgNhP-jmnKYT0N-ZEHGA] [ORM\_HYDRATION\_INSTANCE\_CACHE] [Raw Log](#) Found 0 ArbitraryKeyInd

localhost [12/1 22:29:15] **debug** [W0Bt8zHWbgNhP-jmnKYT0N-ZEHGA] [RAWSQL] [Raw Log](#) Success: 1, Interpolated Query: SELECT `Arbit

[Sat Dec 01 22:29:15 2012] [W0Bt8zHWbgNhP-jmnKYT0N-ZEHGA] [info] [Cache\_Memcached] [0] Memcache host: memcache.shaskins.vm.ny4dev

[Sat Dec 01 22:29:15 2012] [W0Bt8zHWbgNhP-jmnKYT0N-ZEHGA] [info] [Cache\_Memcached] [0] Memcache host: memcache.shaskins.vm.ny4dev

[Sat Dec 01 22:29:15 2012] [W0Bt8zHWbgNhP-jmnKYT0N-ZEHGA] [info] [Cache\_Memcached] [0] Memcache host: memcache.shaskins.vm.ny4dev

[Sat Dec 01 22:29:15 2012] [W0Bt8zHWbgNhP-jmnKYT0N-ZEHGA] [info] [Cache\_Memcached] [0] Memcache host: memcache.shaskins.vm.ny4dev

[Sat Dec 01 22:29:15 2012] [W0Bt8zHWbgNhP-jmnKYT0N-ZEHGA] [info] [Cache\_Memcached] [0] Memcache host: memcache.shaskins.vm.ny4dev

[Sat Dec 01 22:29:15 2012] [W0Bt8zHWbgNhP-jmnKYT0N-ZEHGA] [info] [Cache\_Memcached] [0] Memcache host: memcache.shaskins.vm.ny4dev

[Sat Dec 01 22:29:15 2012] [W0Bt8zHWbgNhP-jmnKYT0N-ZEHGA] [info] [Cache\_Memcached] [0] Memcache host: memcache.shaskins.vm.ny4dev

[Sat Dec 01 22:29:15 2012] [W0Bt8zHWbgNhP-jmnKYT0N-ZEHGA] [info] [Cache\_Memcached] [0] Memcache host: memcache.shaskins.vm.ny4dev

localhost [12/1 22:29:15] **debug** [W0Bt8zHWbgNhP-jmnKYT0N-ZEHGA] [ORM\_HYDRATION\_INSTANCE\_CACHE] [Raw Log](#) Found 0 FeaturedTreasur

localhost [12/1 22:29:15] **debug** [W0Bt8zHWbgNhP-jmnKYT0N-ZEHGA] [RAWSQL] [Raw Log](#) Success: 1, Interpolated Query: SELECT `Featu

[Sat Dec 01 22:29:15 2012] [W0Bt8zHWbgNhP-jmnKYT0N-ZEHGA] [info] [Cache\_Memcached] [0] Memcache host: memcache.shaskins.vm.ny4dev

[Sat Dec 01 22:29:15 2012] [W0Bt8zHWbgNhP-jmnKYT0N-ZEHGA] [info] [Cache\_Memcached] [0] Memcache host: memcache.shaskins.vm.ny4dev



# **Make metrics from logs**

# Big-data analytics

# **Culture of trust**



**Tools solve  
technological friction**

**Culture solves  
organizational friction**

**Get rid of barriers**

# Get rid of barriers

**“Talk to ops to get  
access to the logs”**

# Get rid of barriers

**“Talk to ops to get access to the logs”**

**“Get approval from CTO to push code”**

# Get rid of barriers

**“Talk to ops to get access to the logs”**

**“Get approval from CTO to push code”**

**“Release manager pushes the buttons”**

**Take responsibility**

**Something's broken?**



**Something's broken?**

**Ops problem?**

# Something's broken?

~~Ops problem?~~

**Developer problem?**

# Something's broken?

~~Ops problem?~~

~~Developer problem?~~

**Product problem?**

# Something's broken?

~~Ops problem?~~

~~Developer problem?~~

~~Product problem?~~

# Something's broken?

~~Ops problem?~~

~~Developer problem?~~

~~Product problem?~~

**Problem.**

**Everyone is responsible**

# **Dev/Ops/Product Collaboration**

# **Blameless Postmortems**



# **Why Etsy Loves Continuous Deployment**

**Features get  
developed faster**

# Iterative Development

**Increased confidence**

# **Data-backed decisions**

**Better communication**

# Mean time to recovery

# Tools Mentioned:

- Feature flag tool (not yet released, but soon!)
- Jenkins (<http://jenkins-ci.org>)
- Deployinator (<https://github.com/etsy/deployinator>)
- StatsD (<https://github.com/etsy/statsd>)
- Graphite (<http://graphite.wikidot.com>)
- Ganglia (<http://ganglia.sourceforge.net>)
- Logster (<https://github.com/etsy/logster>)
- Supergrep (<https://github.com/etsy/supergrep>)
- Much more at <https://github.com/etsy>



**Thanks!**