RelEng + QA Quarterly Review

2014 April
Deployment Tooling
- Process through all (useful) pain points from the Dev/Deploy review session
- Scap incremental improvements
  - step 1: Refactor existing scap scripts to enhance maintainability and reveal hidden complexity of current solution
  - step 2: create matrix of tool requirements per software stack (MW, Parsoid, ElasticSearch)
    - Use above matrix to add/fix functionality in scap (or related) tooling for ONE software stack, prioritized by cross stack use
Deployment Tooling

● Process through all (useful) pain points from the Dev/Deploy review session
● Scap incremental improvements
  ○ step 1: Refactor existing scap scripts to enhance maintainability and reveal hidden complexity of current solution
  ○ step 2: Create matrix of tool requirements per software stack (MW, Parsoid, ElasticSearch)
    ■ Use above matrix to add/fix functionality in scap (or related) tooling for ONE software stack, prioritized by cross stack use
Last quarter - What we said

Beta Cluster

- Make database in beta emulate production (set up db slaves)
- Use beta labs as a testing ground for the above Deployment Tooling work
- Migrate Beta cluster from pmtpa to eqiad
Last quarter - What we said

Beta Cluster

- Make database in beta emulate production (set up db slaves)
- Use beta labs as a testing ground for the above Deployment Tooling work
- Migrate Beta cluster from pmtpa to eqiad
  - Much of the beta cluster configuration was puppetized during the migration.
  - Beta now includes a local puppet master which allows cherry-picking work-in-progress puppet changes and applying them across the cluster. This unblocks Antione and others from getting +2 approval in operations/puppet.git for each desired change. It also provides a testing platform for changes prior to usage in production.
  - Beta now includes a salt master which allows the use of Trebuchet and general experimentation with salt by non-roots.
Browser Testing

- use the API to create test data for given tests at run time
- create the ability to test headless
- run versions of tests compatible with target test environments
Browser Testing

- use the API to create test data for given tests at run time
  - first pass: create articles with particular title and content. Create users with particular names and passwords.
- create the ability to test headless
- run versions of tests compatible with target test environments
Last quarter - What we said

Hiring
- Complete hiring and train new Release Engineer
- Complete hiring and train new Automation Engineer (Ruby)
Last quarter - What we said

Hiring

- Complete hiring and train new Release Engineer
- Complete hiring and train new Automation Engineer (Ruby)
Next quarter (diff)
Make the tool that deploys in production the tool that deploys in Beta Cluster (May-Jun)

** Improvements to scap

** Complete and holistic view of running tests:

** feature coverage

*** Cucumber Scenarios test descriptions created and maintained according to demands of feature teams (not invented by QA staff)

** build time and overhead

*** use headless Firefox builds in WMF Jenkins for fast functional testing

*** tests use API to manage test data in arbitrary target wikis

*** use branch tags to run tests appropriate to the builds being tested

** system diagnostics

*** Add more reporting and analysis tools such as fatalities monitoring, icinga, logstash, detailed error reporting from Jenkins

** Browser testing underway with all engineering teams

* 2014<br />

[[Wikimedia_Release_and_QA_Team/Quarterly_review_April_2014|review]]

[[Deployment tooling]]:

* integrate HHVM support into our deployment system

* start the scap(p)y & trebuchet integration conversation

* Complete transition to scap as code deploy system

* Support HHVM deployment tooling and puppet configuration testing

Mediawiki Release:

* Successfully support the release of MediaWiki 1.23

QA:

* Retire Cloudbees Jenkins instance

* Integrate WMF Jenkins with new WMF SauceLabs account

* Execute tests in parallel

* Use tags to run builds appropriate to released versions (e.g. don't run master build on fast2wiki)

* Use API to create test data at runtime more widely (not just for MobileFrontend but also VisualEditor, Flow, local dev env etc.)

* Add browsertests to new repos e.g. GettingStarted

Hiring:

Next quarter (What we now plan)
Next quarter

Deployment tooling

● (continued from last quarter) Process through all (useful) pain points from the Dev/Deploy review session
● Integrate HHVM support into our deployment system
● start the scap(py) & trebuchet integration conversation
Next quarter

Beta cluster

- Complete transition to scap as code deploy system
- Support HHVM deployment tooling and puppet configuration testing
- Swift cluster in beta (stretch goal)
Next quarter

MediaWiki Release

- Successfully support the release of MediaWiki 1.23
- Kick off/complete second RFP
Next quarter

QA

- (From last quarter) Use tags to run builds appropriate to released versions (e.g. don't run master build on test2wiki)
- (continued) Retire Cloudbees Jenkins instance
- Integrate WMF Jenkins with new WMF SauceLabs account
  - Execute tests in parallel
- Use API to create test data at runtime more widely (not just for MobileFrontend but also VisualEditor, Flow, local dev env etc.)
- Add browsertests to new repos e.g. GettingStarted
Next quarter

Hiring

- Complete hiring and train new Release Engineer and Automation Engineer (Ruby)
Next quarter - Dependencies

Ops:
● Swift
● Icinga (continued)
Next quarter - Dependencies

MW Core:
- hhvm
Questions

(if you haven’t already asked them)