

CONTINUOUS DEPLOYMENT AND DEVOPS

D E P R E C A T I N G S I L O S

JOSH DEVINS, NOKIA
TOM SULSTON, THOUGHTWORKS

JAOO 2010
ÅRHUS, DENMARK

WHO ARE WE AND WHERE ARE WE FROM?

- Josh Devins, Nokia Berlin
 - Software architect, Location Services
 - Sysadmin of honour
- Tom Sulston, ThoughtWorks
 - Lead consultant
 - DevOps, build & deploy



PROBLEM SITUATION

Monday, October 4, 2010

3

A few words of introduction on what the “before” state was

- web and device
- growth from startup to millions of devices/mo
- free navigation earlier this year increased usage
- rapid feature and team growth



DEVELOPMENT AND OPERATIONS SILOS

Monday, October 4, 2010

4

<http://www.flickr.com/photos/tonyjcase/4092410854/sizes/l/in/photostream/>

Developers and operations teams separated both organisationally and physically
Whole different organisational structure – need to go to C-level (VP-level?) to find a common reporting line

Started as a hardware company, and really bolted on services at the beginning

Poor alignment of technology choices (base OS, packaging, monitoring)

Very little common ground, because...



MANY SEPARATE TEAMS

Monday, October 4, 2010

5

- lots of technology/approach divergence caused by:
- many ops teams - “operations”, “transitions”, “development support”
- many development teams - frontend, backend, backend function x/y/z
 - Conway’s Law
 - short term scaled well and fast
 - right intention of giving small teams autonomy but...balance needed
- Lots of integration points
 - more complexity than necessary
 - lots of inventory
 - Integration is v. painful



TOO MUCH MANUAL WORK

Monday, October 4, 2010

6

- lots of things done by hand, non-repeatable
 - QA, almost nothing automated (except where really necessary -- perf tests)
 - Baroque configuration process
 - Releases take a long time and a lot of manual testing/verification
 - Cycle time is very slow
 - Right intentions, did not scale
 - change management process (?)
 - carrying knowledge/understanding across silos has a cost (x4)
- Frequent rework – fixing the same problem again and again and usually at the last-minute



Monday, October 4, 2010

7

<http://www.flickr.com/photos/14608834@N00/2260818367/sizes/o/in/photostream/>

- reality: about one and a half people knew how the whole thing worked end-to-end
- reality: ~10-days to build a new image with Java, 5 Tomcat instances, as many war files, nothing else!
- worse: the "image system" was not used anywhere except staging and production so failures can very late
- maintenance: in dev/QA regular Debian systems with DEB packaging was used, had to essentially maintain two complete distribution mechanisms
- change management process is heavyweight
 - ITIL++, multi-tab Excel spreadsheets, CABs in other countries, not directly involved
 - often circumvented
 - communication gaps between ops teams
- package and config structure (ISO + rsync)
- it worked, but was slow and cryptic
- building whole OS images in very slow and non-parallelisable (4 hrs?) CI
- multi-phased approach requiring first a custom packaging system and description language (VERY cryptic and bespoke)
- using PXE Linux to boot images from a central control server for configuration rsync
- any booted server can act as a peer to boot other machines

A photograph showing a white power strip mounted on a wall. Several power adapters are plugged into it, including one labeled 'VCR', 'HIYAMA', 'PRI-INTER', and 'WACOM'. A '4WAY EXT.' adapter is also visible. A large, tangled mess of black and white cables hangs from the strip and extends across the frame, illustrating a disorganized infrastructure.

AD-HOC INFRASTRUCTURE MANAGEMENT

Monday, October 4, 2010

8

<http://www.flickr.com/photos/14608834@N00/2260818367/sizes/o/in/photostream/>

- lots of things done by hand, non-repeatable
- “We don’t have time to do it right”
- time-to-recovery is slow
- monitoring is:
 - inconsistent (lots of false alarms)
 - unclear (multiple tools, teams)
 - too coarse (the site is down!)
- hard to triage infrastructure or code issues
- inventory management is weak
- many data centres,
- not enough knowledge kept in-house



MAKING IT BETTER

Monday, October 4, 2010

9

- Any questions on describing the problem?
 - has anyone got similar problems?
- What actions did we take to address these issues?

Time check: 20 mins

CONTINUOUS DELIVERY



Monday, October 4, 2010

10

<http://www.flickr.com/photos/snogging/4688579468/sizes/l/>

- what is continuous delivery?
- Continuous Delivery: every SCM commit results in releasable software
- that is, from a purely infrastructural and "binary-level" perspective, the software is always releasable
- This includes layers of testing, not just releasing anything that compiles!
- features may be incomplete, etc. so in practice you might not actually release every commit (ie: Continuous Deployment)
- "If something hurts, do it more often"
- You should have gone to Jez's session this morning!

CONTINUOUS DELIVERY



More!

Monday, October 4, 2010

10

<http://www.flickr.com/photos/snogging/4688579468/sizes/l/>

- what is continuous delivery?
- Continuous Delivery: every SCM commit results in releasable software
- that is, from a purely infrastructural and "binary-level" perspective, the software is always releasable
- This includes layers of testing, not just releasing anything that compiles!
- features may be incomplete, etc. so in practice you might not actually release every commit (ie: Continuous Deployment)
- "If something hurts, do it more often"
- You should have gone to Jez's session this morning!



CONTINUOUS INTEGRATION AND BUILD PIPELINE

Monday, October 4, 2010

11

http://www.uvm.edu/~wbowden/Image_files/Pipeline_at_Kuparuk.jpg

- how do we get from a SCM commit to something that is deployable and tested enough?
- Building the 'conveyor belt'
- Turn up existing CI practices to 11
 - Each team already did "build & unit test" - no deployable package (WARs to Nexus)
 - Automated integration of various teams' work
 - Automated integration testing
 - Testing deployments - same method on all environments
- Currently using Hudson & ant - this works OK.



More!

CONTINUOUS INTEGRATION AND BUILD PIPELINE

Monday, October 4, 2010

11

http://www.uvm.edu/~wbowden/Image_files/Pipeline_at_Kuparuk.jpg

- how do we get from a SCM commit to something that is deployable and tested enough?
- Building the 'conveyor belt'
- Turn up existing CI practices to 11
 - Each team already did "build & unit test" - no deployable package (WARs to Nexus)
 - Automated integration of various teams' work
 - Automated integration testing
 - Testing deployments - same method on all environments
- Currently using Hudson & ant - this works OK.



A DIVERSION INTO MAVEN PAIN

Monday, October 4, 2010

12

<http://www.petsincasts.com/?p=162>

- workaround: don't use the Maven "release" process or just live with it and do Maven "releases" as often as possible
- lesson learned: don't try to mess with "the Maven way", it gets very hairy and is a huge time suck
- lesson learned: don't depend on SNAPSHOT dependencies unless they are under your own control (can't safely release your module with SNAPSHOT deps meaning you will have to wait for someone else to release their module)
- standard Maven versioning lifecycle: 1.0.0-SNAPSHOT, pull down dependencies (some SNAPSHOTs themselves) from some repository (usually one that is not integrated with your source code repository)
- working away on 1.0.0-SNAPSHOT and I'm ready to release so then do a Maven "release", tagging SCM, and I get version 1.0.0
- crap we found a bug, so we keep working now on version 1.0.1-SNAPSHOT
- okay, ready to release again so I get version 1.0.1
- do some testing and everything is happy so I drop my 1.0.1 war into my production Tomcat
- what's wrong with this picture?
- key: we "release" software BEFORE we are satisfied with its' quality
- like we said before, continuous delivery is all about the possibility of releasing to production at all times, from all commits



Less!

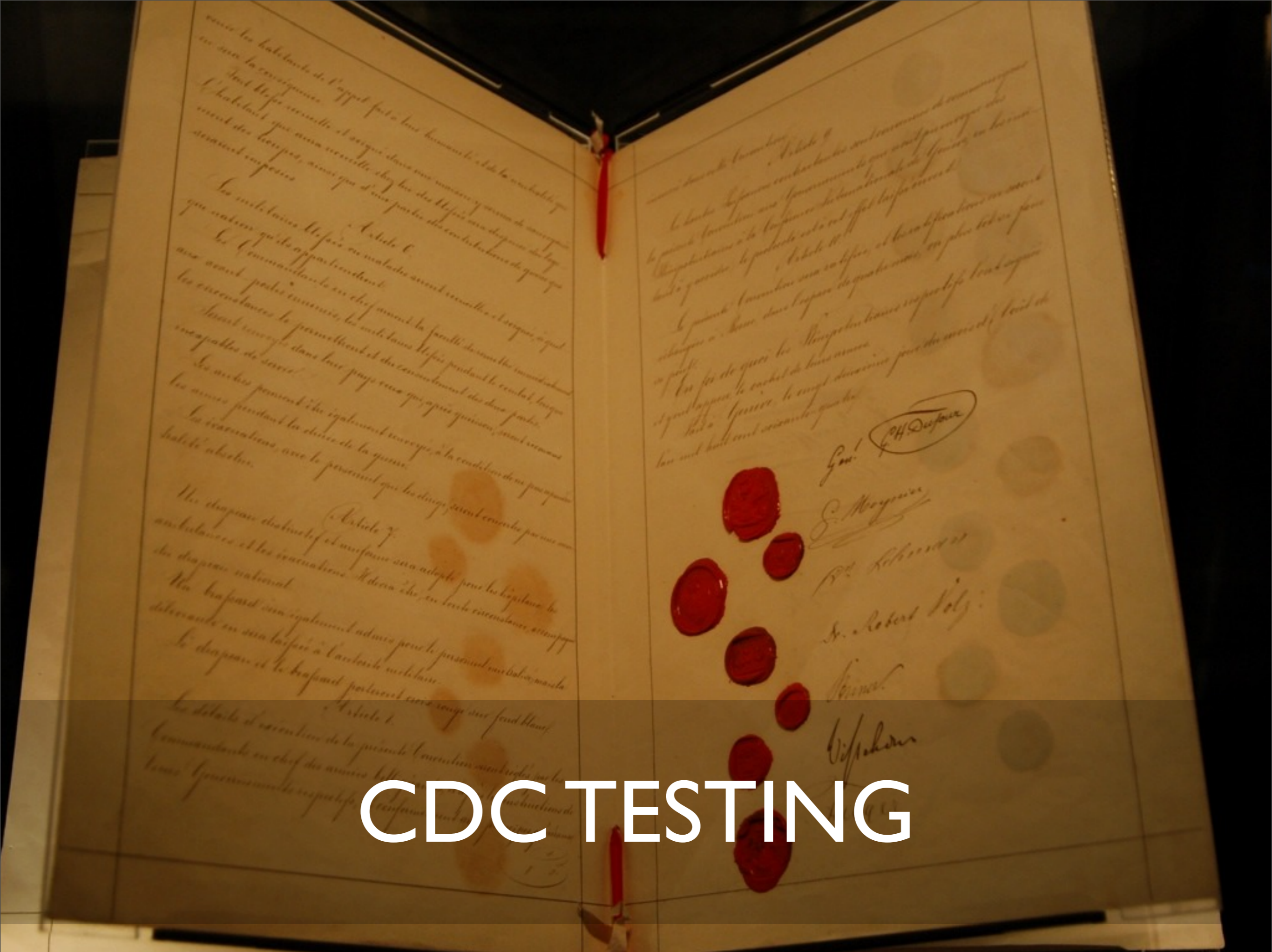
A DIVERSION INTO MAVEN PAIN

Monday, October 4, 2010

12

<http://www.petsincasts.com/?p=162>

- workaround: don't use the Maven "release" process or just live with it and do Maven "releases" as often as possible
- lesson learned: don't try to mess with "the Maven way", it gets very hairy and is a huge time suck
- lesson learned: don't depend on SNAPSHOT dependencies unless they are under your own control (can't safely release your module with SNAPSHOT deps meaning you will have to wait for someone else to release their module)
- standard Maven versioning lifecycle: 1.0.0-SNAPSHOT, pull down dependencies (some SNAPSHOTs themselves) from some repository (usually one that is not integrated with your source code repository)
- working away on 1.0.0-SNAPSHOT and I'm ready to release so then do a Maven "release", tagging SCM, and I get version 1.0.0
- crap we found a bug, so we keep working now on version 1.0.1-SNAPSHOT
- okay, ready to release again so I get version 1.0.1
- do some testing and everything is happy so I drop my 1.0.1 war into my production Tomcat
- what's wrong with this picture?
- key: we "release" software BEFORE we are satisfied with its' quality
- like we said before, continuous delivery is all about the possibility of releasing to production at all times, from all commits



CDC TESTING

Monday, October 4, 2010

13

CDC – Consumer-Driven Contract

<http://www.martinfowler.com/articles/consumerDrivenContracts.html>

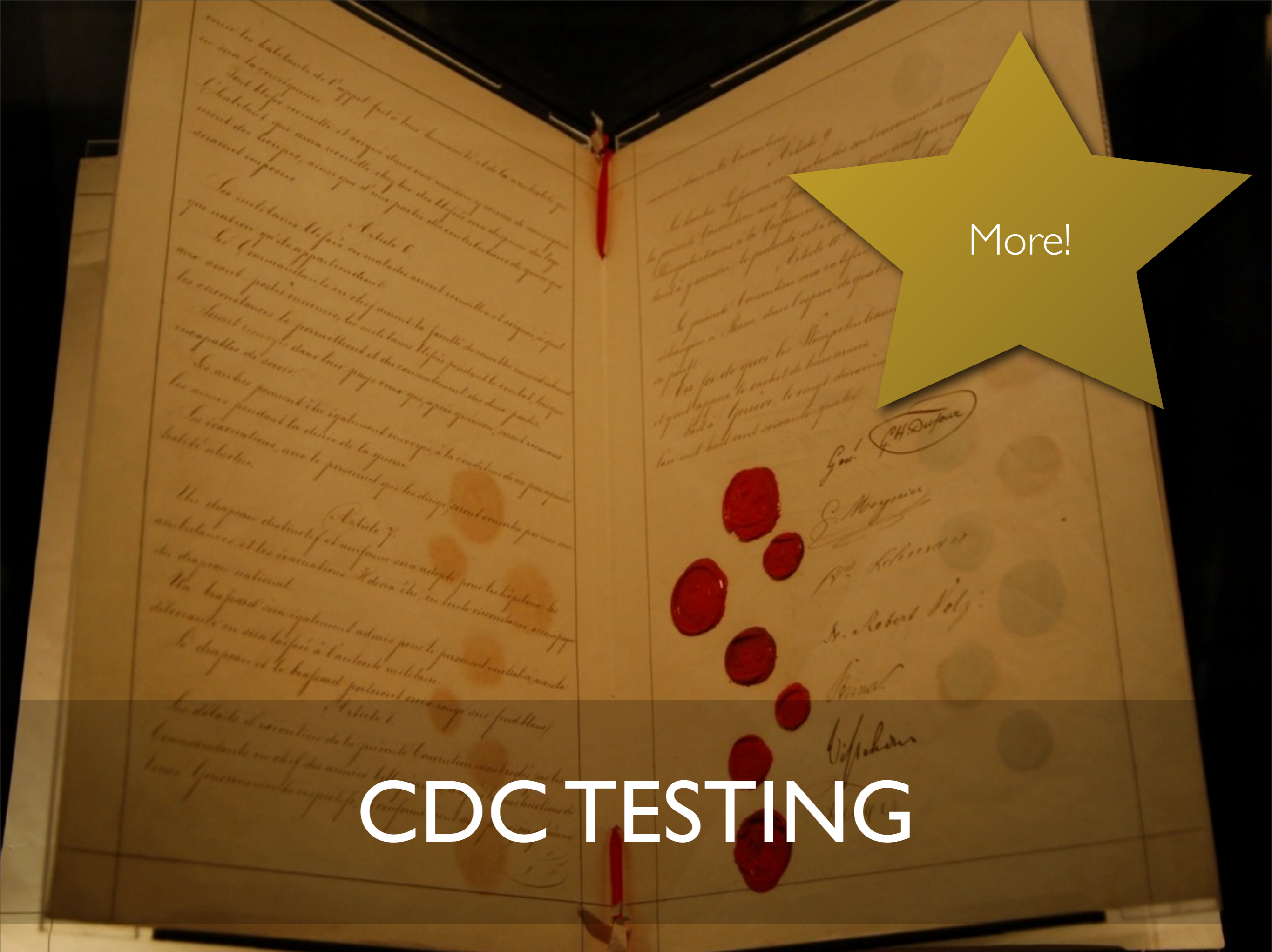
Each service/team provides tests for those teams whose services they consume. (ie: If I use your service, I write you a test that expresses how I am using it. You can then run that test in your build.)

Lets us do quick integration-type testing at the unit/functional level.

Much easier than maintaining stubs.

Designed to catch integration failures earlier (typical failure mode is for clients/servers to diverge while still passing their own tests, only to be caught at manual QA stages)

Ceremony for giving tests to another team



More!

CDC TESTING

Monday, October 4, 2010

13

CDC – Consumer-Driven Contract

<http://www.martinfowler.com/articles/consumerDrivenContracts.html>

Each service/team provides tests for those teams whose services they consume. (ie: If I use your service, I write you a test that expresses how I am using it. You can then run that test in your build.)

Lets us do quick integration-type testing at the unit/functional level.

Much easier than maintaining stubs.

Designed to catch integration failures earlier (typical failure mode is for clients/servers to diverge while still passing their own tests, only to be caught at manual QA stages)

Ceremony for giving tests to another team



PACKAGING: RPM & YUM

Monday, October 4, 2010

14

<http://www.flickr.com/photos/delgrossodotcom/2553424895/>

- Build once!
- passing deployable packages (RPMs) up the value chain
 - Categorically 100% sure that you're testing what you're going to deploy
- Can wrap up all sorts of useful things in OS packages
 - reference data
 - hook scripts
 - dependencies on tiered applications
- build pipeline of repositories
 - Each repo means "X level of testing has been done on these packages"
 - gotcha: createrepo caching
 - gotcha: no concurrent running of createrepo
 - gotcha: using metapackages to join versions (Might re-introduce in future)



Keep doing!

PACKAGING: RPM & YUM

Monday, October 4, 2010

14

<http://www.flickr.com/photos/delgrossodotcom/2553424895/>

- Build once!
- passing deployable packages (RPMs) up the value chain
 - Categorically 100% sure that you're testing what you're going to deploy
- Can wrap up all sorts of useful things in OS packages
 - reference data
 - hook scripts
 - dependencies on tiered applications
- build pipeline of repositories
 - Each repo means "X level of testing has been done on these packages"
 - gotcha: createrepo caching
 - gotcha: no concurrent running of createrepo
 - gotcha: using metapackages to join versions (Might re-introduce in future)



RDBMS, NOSQL, DATA DEPLOYMENT

Monday, October 4, 2010

15

- not doing this yet, but here are some ideas
- Currently using MySQL – is there a need to change to Key/Value store?
- RDBMS: check out ???
- NoSQL: big, huge question mark and little tooling support, so consider this seriously if considering NoSQL
- some teams are using BitTorrent to distribute large (GB and TB) datasets around the world – Lucene indices, map files, etc.
- similar to the idea that Twitter uses to deploy stuff with their Murder tool
- can we use dbdeploy?



???

RDBMS, NOSQL, DATA DEPLOYMENT

Monday, October 4, 2010

15

- not doing this yet, but here are some ideas
- Currently using mySQL – is there a need to change to Key/Value store?
- RDBMS: check out ???
- NoSQL: big, huge question mark and little tooling support, so consider this seriously if considering NoSQL
- some teams are using BitTorrent to distribute large (GB and TB) datasets around the world – Lucene indices, map files, etc.
- similar to the idea that Twitter uses to deploy stuff with their Murder tool
- can we use dbdeploy?

PUPPET



Monday, October 4, 2010

16

- Puppet overview & alternatives (Chef, CFEngine, hand-rolled tools)
- manifests
- modules and inheritance
- passing puppet configs with deployable code + configs
- Driven from developer-facing sysadmins



More!

PUPPET



Monday, October 4, 2010

16

- Puppet overview & alternatives (Chef, CFEngine, hand-rolled tools)
- manifests
- modules and inheritance
- passing puppet configs with deployable code + configs
- Driven from developer-facing sysadmins



BDD

Monday, October 4, 2010

17

- infrastructure testing with cucumber-puppet
- applying good development practices to the Ops world
- absolutely crucial to having a refactorable infrastructure
 - how unchanging are your systems?
- can we start doing Behaviour-driven releases?
- This is alpha software!
- Does not catch all errors

A close-up photograph of a green cucumber against a black background. A gold-colored five-pointed star is superimposed on the left side of the cucumber. The word "More!" is written in white text inside the star.

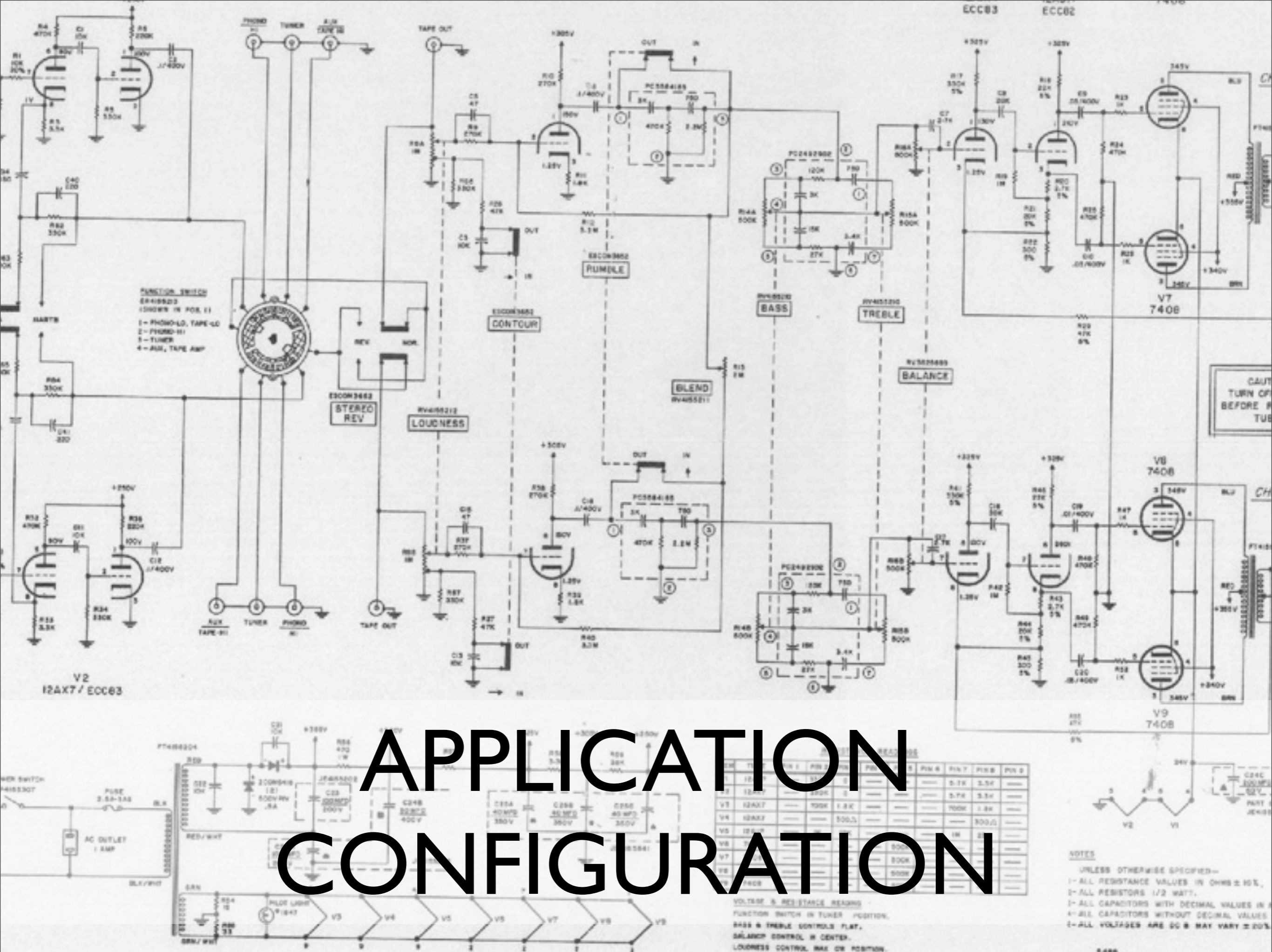
More!

BDD

Monday, October 4, 2010

17

- infrastructure testing with cucumber-puppet
- applying good development practices to the Ops world
- absolutely crucial to having a refactorable infrastructure
 - how unchanging are your systems?
- can we start doing Behaviour-driven releases?
- This is alpha software!
- Does not catch all errors

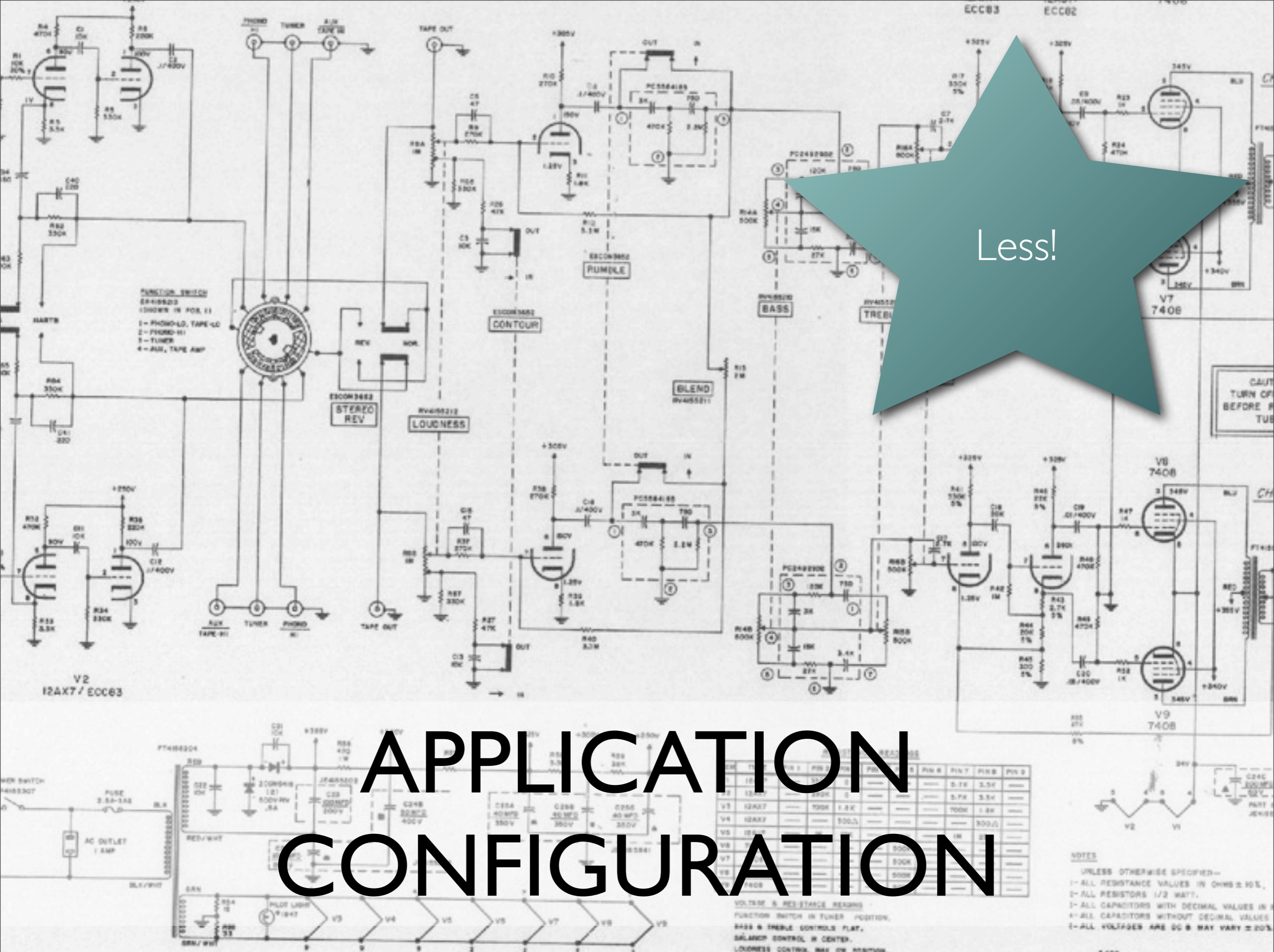


APPLICATION CONFIGURATION

Monday, October 4, 2010

18

- Configurations passed up from development team through Subversion
- Deployed with puppet
- Tested with cucumber-puppet
- Tested on application start for missing values
- Bundling application deployments simplifies configuration
- TODO: review architecture of all apps and simplify (easier now that deployment tech debt is reduced)



Monday, October 4, 2010

18

Configurations passed up from development team through Subversion
Deployed with puppet
Tested with cucumber-puppet
Tested on application start for missing values
Bundling application deployments simplifies configuration
TODO: review architecture of all apps and simplify (easier now that deployment tech debt is reduced)



PRE-FLIGHT TESTING

Monday, October 4, 2010

19

<http://www.flickr.com/photos/jimbl/2881681649/sizes/o/>

- scripted checks before anything even happens
- ensure that the stage is set and all known pre-requisites are tested and monitored
- application health-check on startup (are all my config values set?)
- check_http through nrpe



PRE-FLIGHT TESTING

Monday, October 4, 2010

19

<http://www.flickr.com/photos/jimbl/2881681649/sizes/o/>

- scripted checks before anything even happens
- ensure that the stage is set and all known pre-requisites are tested and monitored
- application health-check on startup (are all my config values set?)
- check_http through nrpe



Monday, October 4, 2010

20

<http://www.flickr.com/photos/kylesteddesign/4395772305/sizes/o/>

- speaking of monitoring...
- Nagios, nrpe
- cucumber-nagios – Monitoring-driven deployments?

Would like developers to push up monitors alongside features.

- developers and engineers gaining common understanding around monitoring and system behaviour



MONITORING

Monday, October 4, 2010

20

<http://www.flickr.com/photos/kylesteddesign/4395772305/sizes/o/>

- speaking of monitoring...
- Nagios, nrpe
- cucumber-nagios – Monitoring-driven deployments?

Would like developers to push up monitors alongside features.

- developers and engineers gaining common understanding around monitoring and system behaviour



ITIL, DEVOPS AND YOU

Monday, October 4, 2010

21

ITIL is a framework. DevOps is a series of practices.
While you could have lightweight ITIL implementations, they tend to be process-heavy.
DevOps is about doing all the good technical diligence in a way that marries with Agile practices and values
– not dependent on tool choice
Build up shared understanding by automation
Jez: A document proves nothing. But a script is real proof that you have done what is in the script.



More
automation

ITIL, DEVOPS AND YOU

Monday, October 4, 2010

21

ITIL is a framework. DevOps is a series of practices.
While you could have lightweight ITIL implementations, they tend to be process-heavy.
DevOps is about doing all the good technical diligence in a way that marries with Agile practices and values
– not dependent on tool choice
Build up shared understanding by automation
Jez: A document proves nothing. But a script is real proof that you have done what is in the script.



Less
administration

More
automation

ITIL, DEVOPS AND YOU

Monday, October 4, 2010

21

ITIL is a framework. DevOps is a series of practices.

While you could have lightweight ITIL implementations, they tend to be process-heavy. DevOps is about doing all the good technical diligence in a way that marries with Agile practices and values

- not dependent on tool choice

Build up shared understanding by automation

Jez: A document proves nothing. But a script is real proof that you have done what is in the script.



WHERE ARE WE?

Monday, October 4, 2010

22

- not doing continuous deployment, but are making-ready
- it takes time for large organisations to catch up to technical change
- addressing cultural issues
- building common understanding and shared ownership

JOIN US!

- Nokia is hiring in Berlin!
 - www.nokia.com/careers
- ThoughtWorks is hiring in London, Hamburg and further abroad.
 - www.thoughtworks.com/jobs

THANKS!

JOSH DEVINS

www.joshdevins.net

@joshdevins

TOM SULSTON

www.thoughtworks.com

@tomsulston

JOSH DEVINS, NOKIA
TOM SULSTON, THOUGHTWORKS

JAOO 2010
ÅRHUS, DENMARK