From Dev to DevOps

Carlos Sanchez
@csanchez

http://carlossanchez.eu

@csanchez

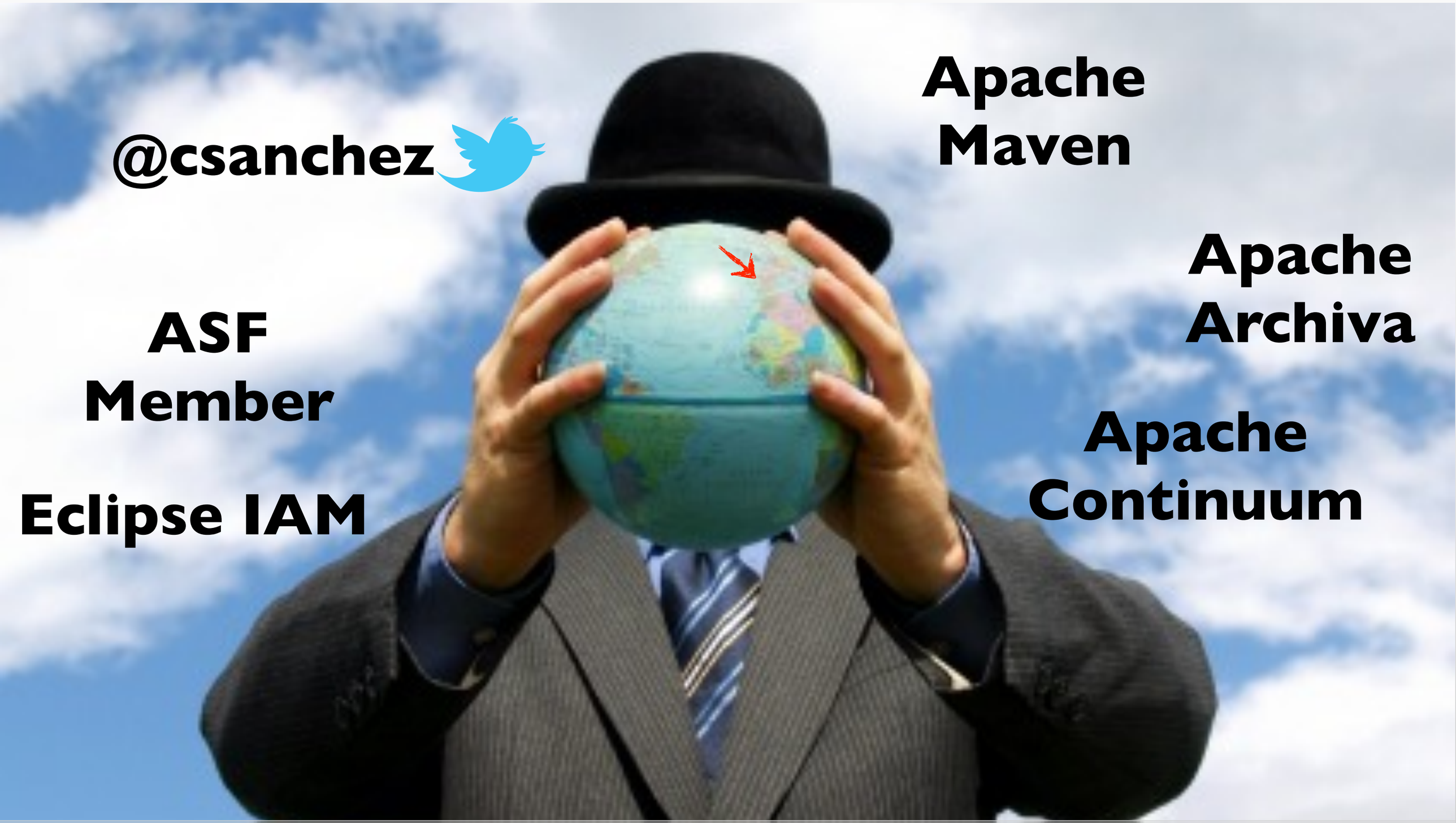
Apache Maven

Apache Archiva

ASF Member

Apache Continuum

Eclipse IAM



Dev... What?

Agile

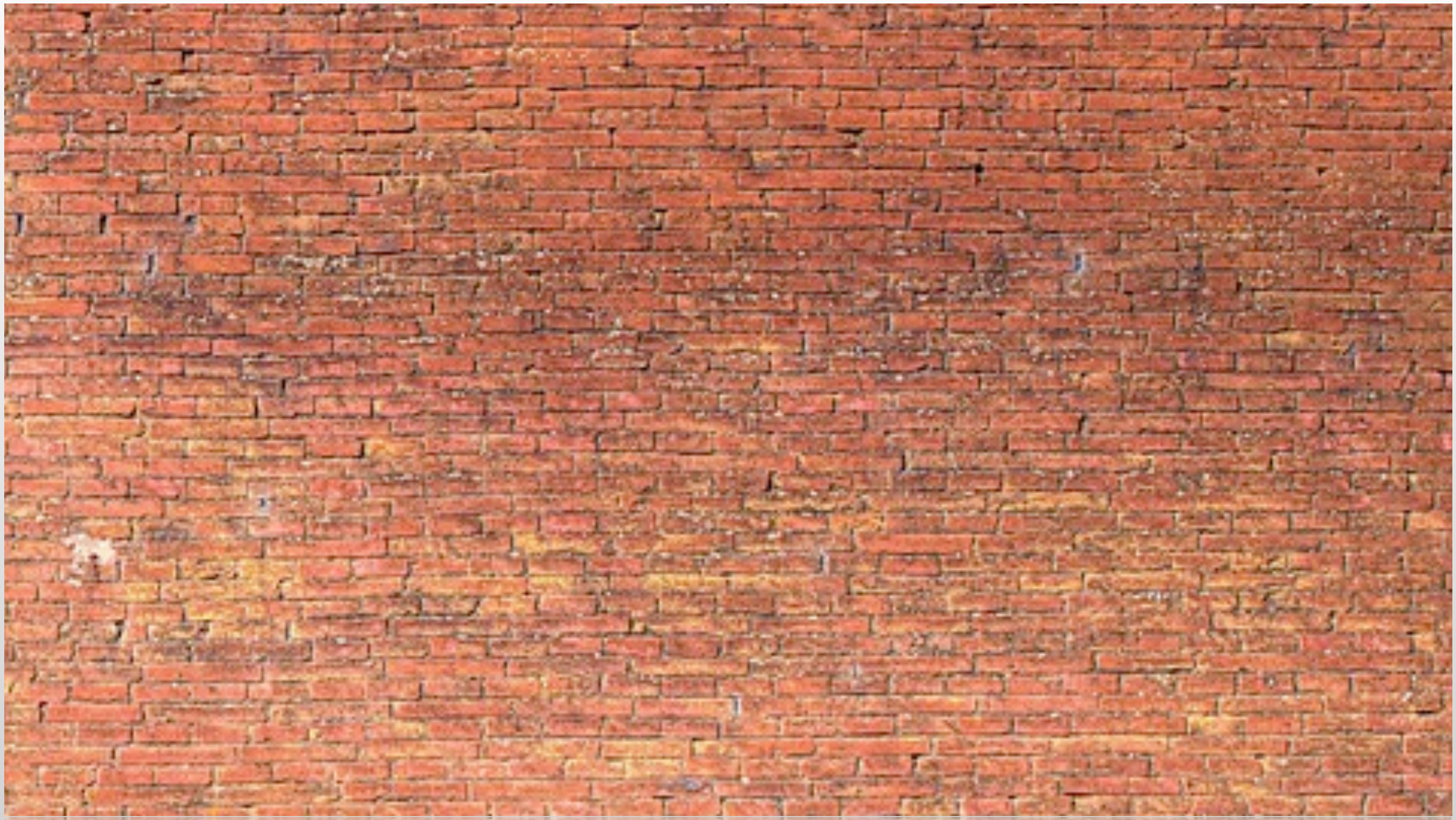
planning

iterative development

continuous integration

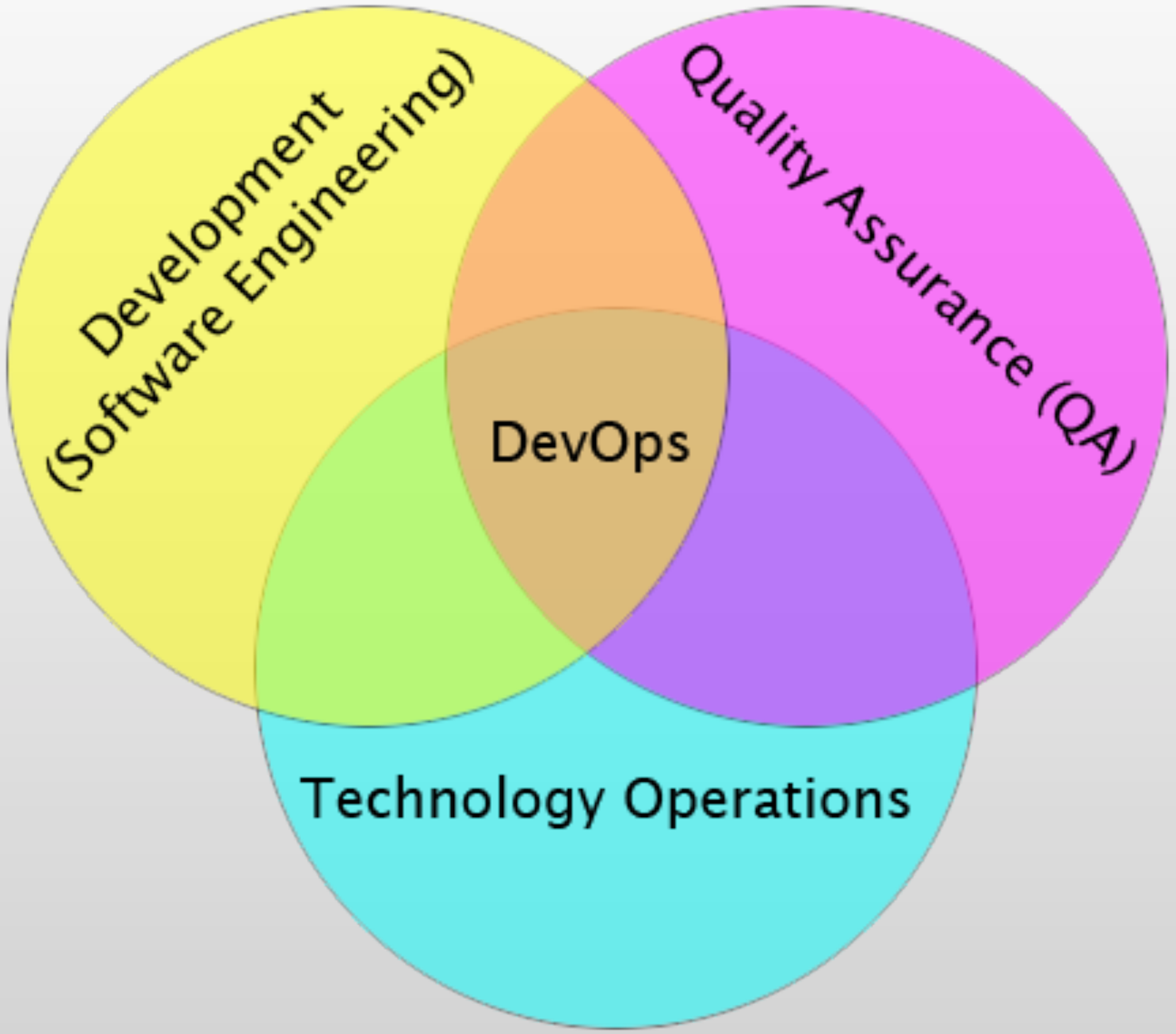
release soon, release often

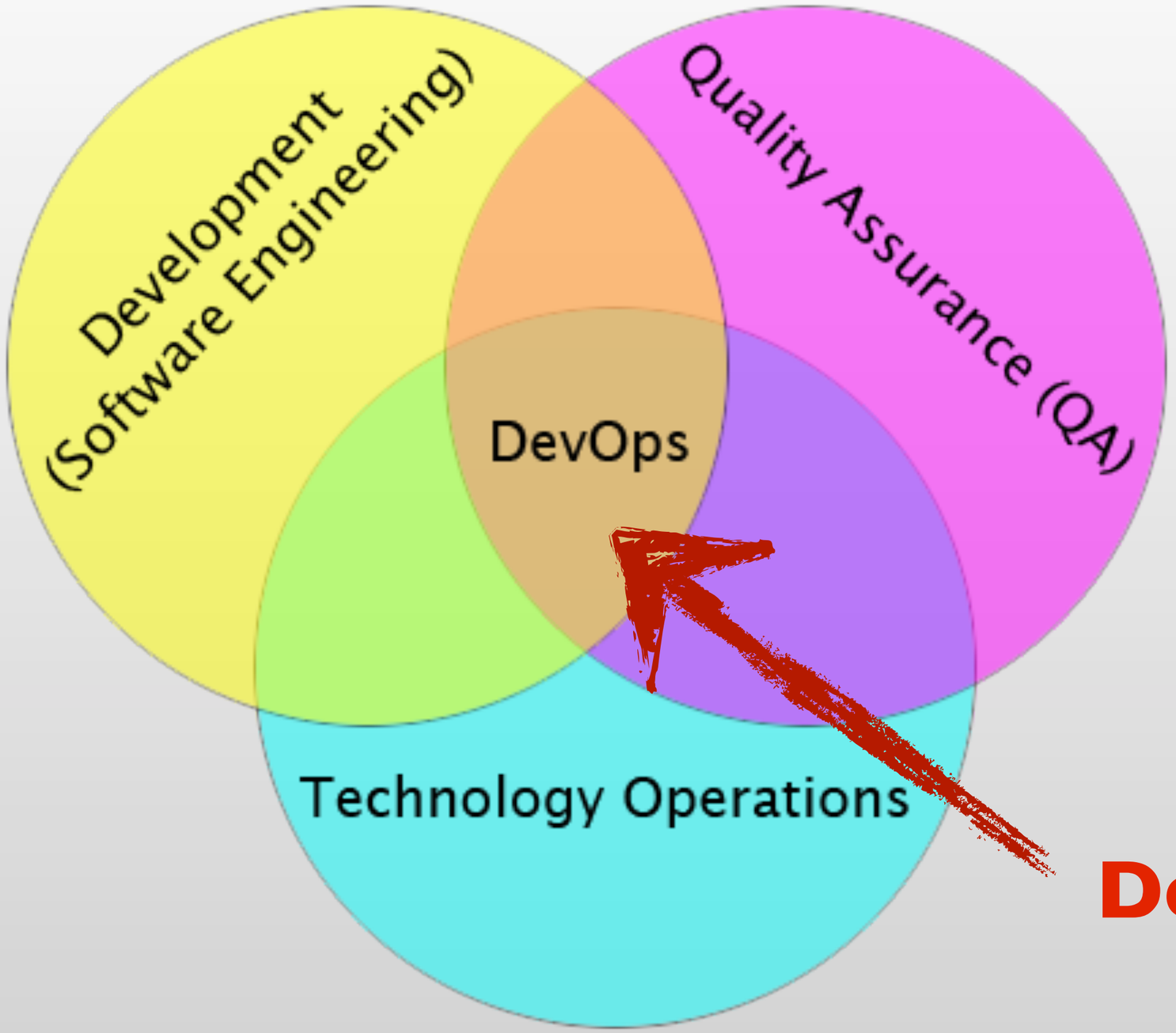

Apache
Con
North America
2011



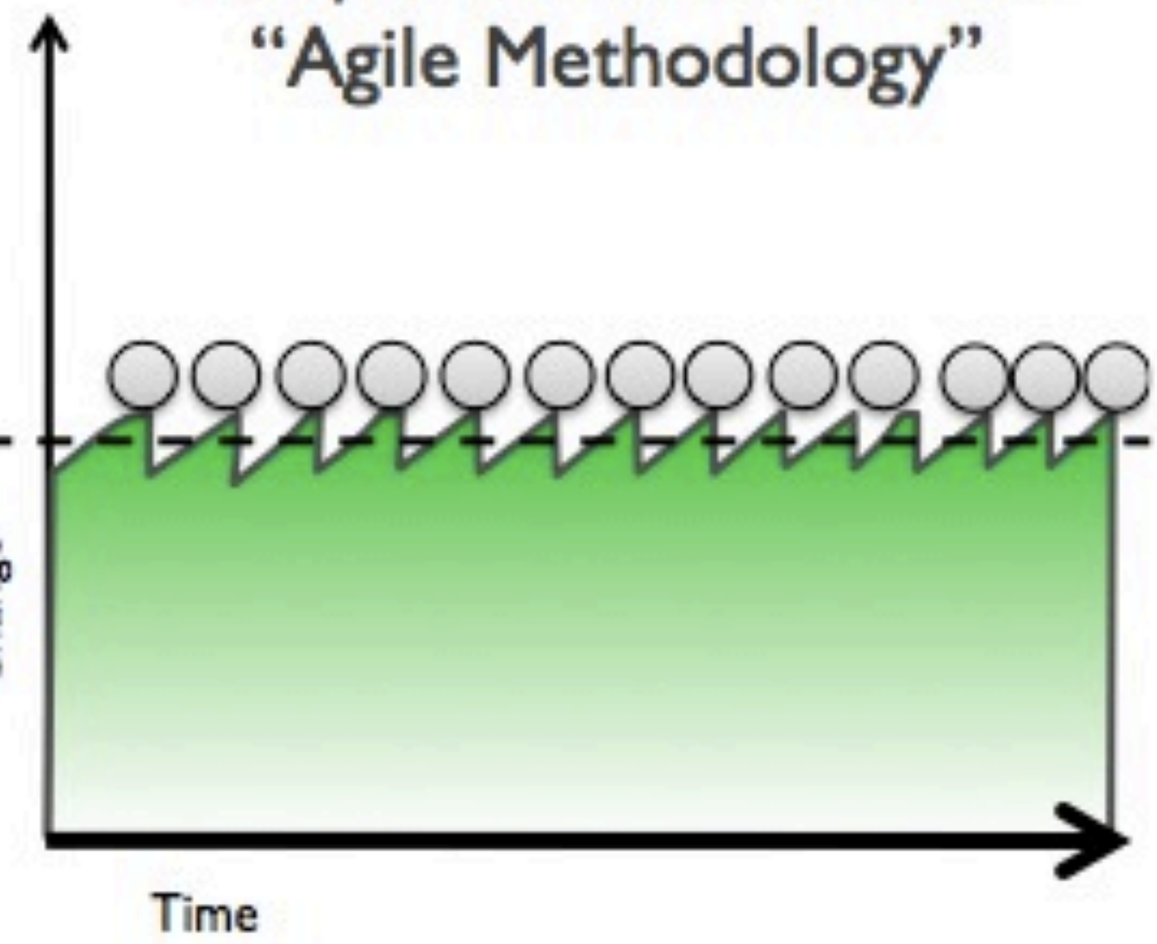
Apache Con North America 2011

Apache Con North America 2011


Development
(Software Engineering)
Quality Assurance (QA)
DevOps
Technology Operations

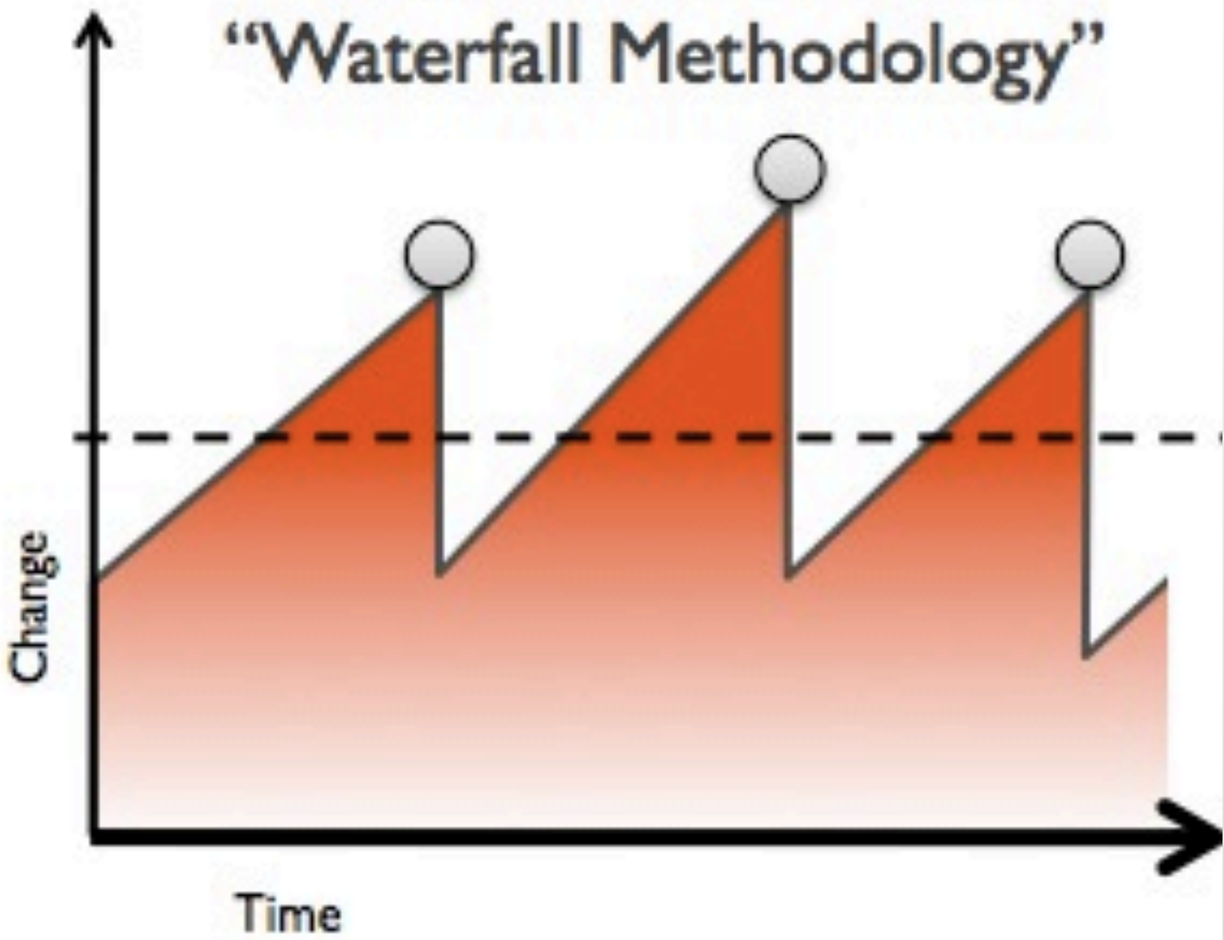

Development
(Software Engineering)
Quality Assurance (QA)
DevOps
Technology Operations

DevQaOps ?



Smoother Effort

Less Risk



Effort Peaks

High Risk

DevOps addresses

Fear of change

Risky deployments

It works on my machine!

Siloisation

Dev Change vs. Ops stability

Individuals and interactions over processes and tools

Working software over comprehensive documentation

Customer collaboration over contract negotiation

Responding to change over following a plan

Our highest priority is to satisfy the customer through early and continuous delivery of valuable **software**.

Welcome changing requirements, even **late in development**. Agile processes harness change for the customer's competitive advantage.

Deliver **working software** frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

Business people and **developers** must work together daily throughout the project.

The most efficient and effective method of conveying information to and within a **development team** is face-to-face conversation.

Agile processes promote sustainable **development**. The **sponsors, developers, and users** should be able to maintain a constant pace indefinitely.

A Beautiful Mind

"An even
greater gift is to discover
a beautiful heart."

RUSSELL CROWE ~ ED HARRIS ~ JENNIFER CONNELLY

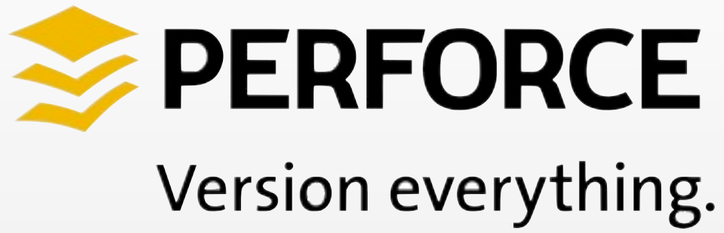
Dev


Apache
Con
North America
2011


GIT


mercurial

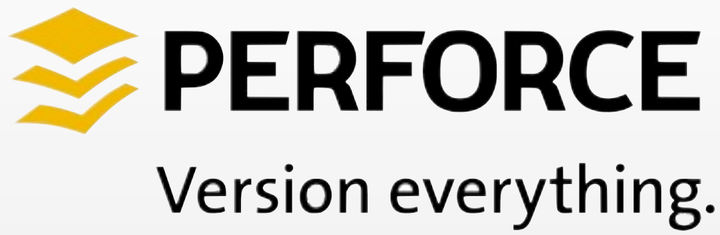

SUBVERSION


PERFORCE
Version everything.


GIT


mercurial


SUBVERSION


PERFORCE
Version everything.


buildr


ivy

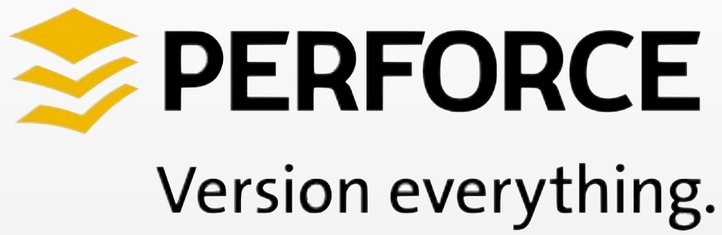

maven


Apache
Con
North America
2011


GIT


mercurial


SUBVERSION


PERFORCE
Version everything.


buildr


ivy


maven


continuum


Apache
Con
North America
2011

GIT

mercurial

SUBVERSION

PERFORCE
Version everything.

buildr

ivy

Maven

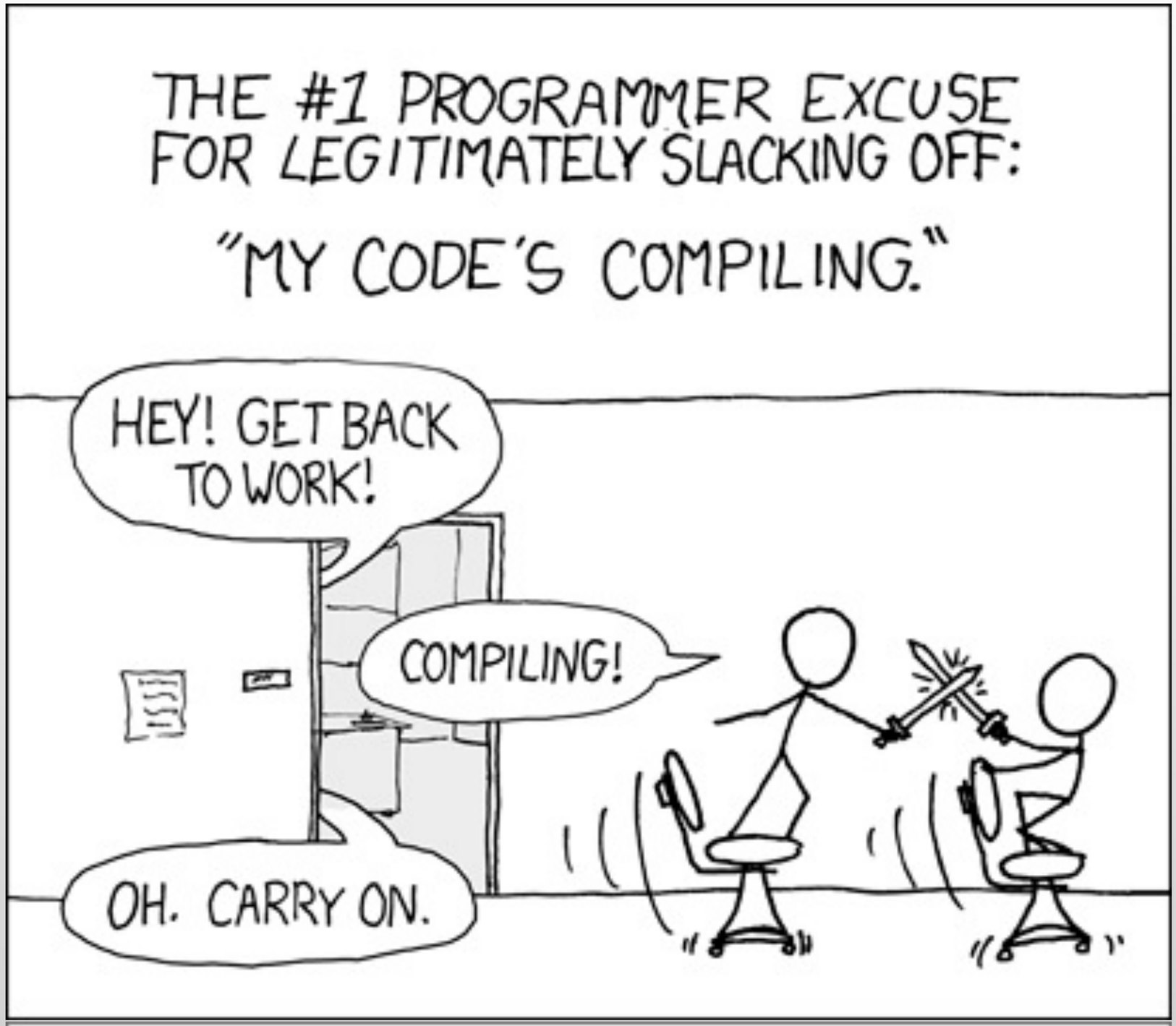
continuum

Nexus

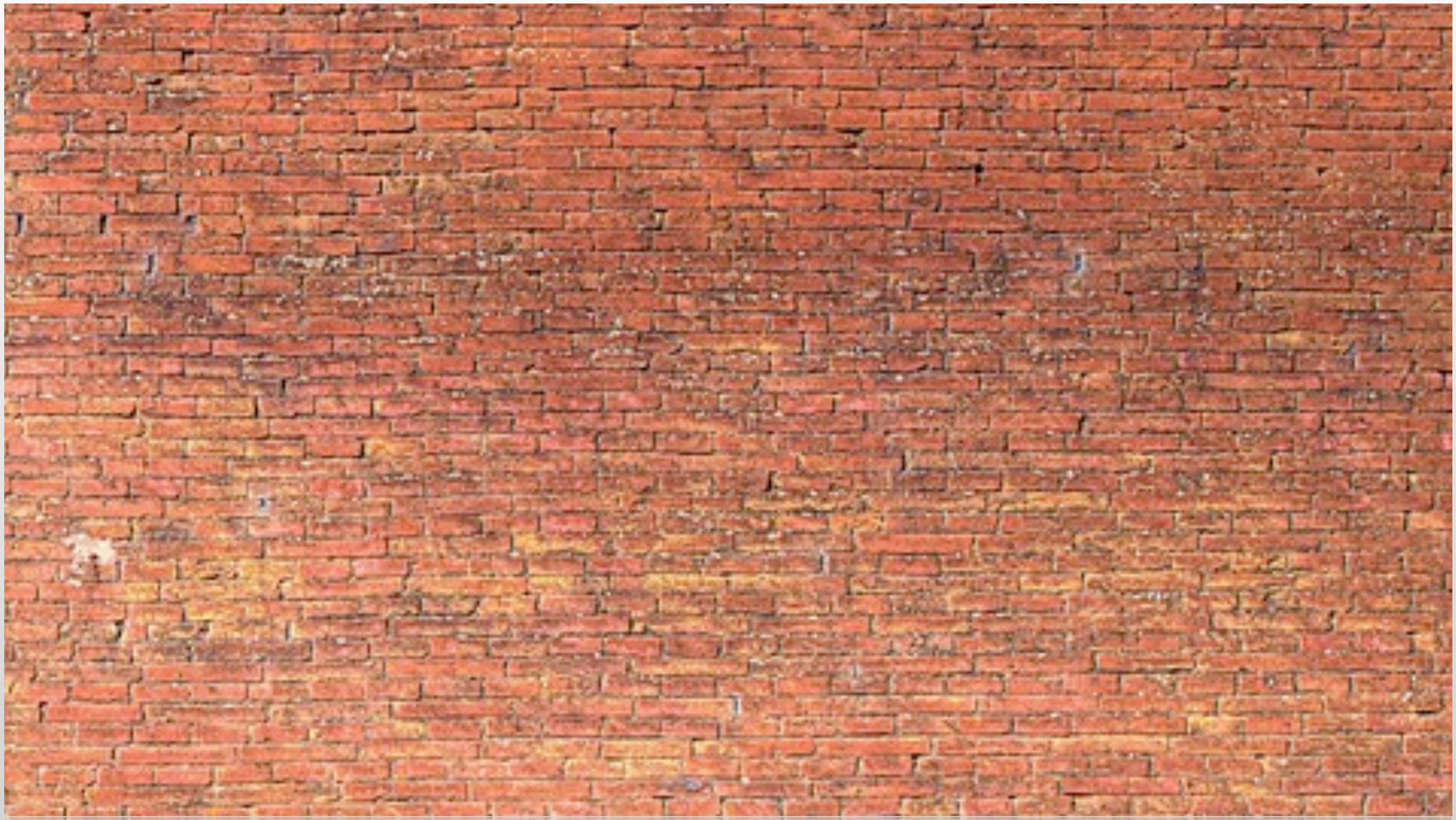
archiva

artifactory

Apache Con North America 2011

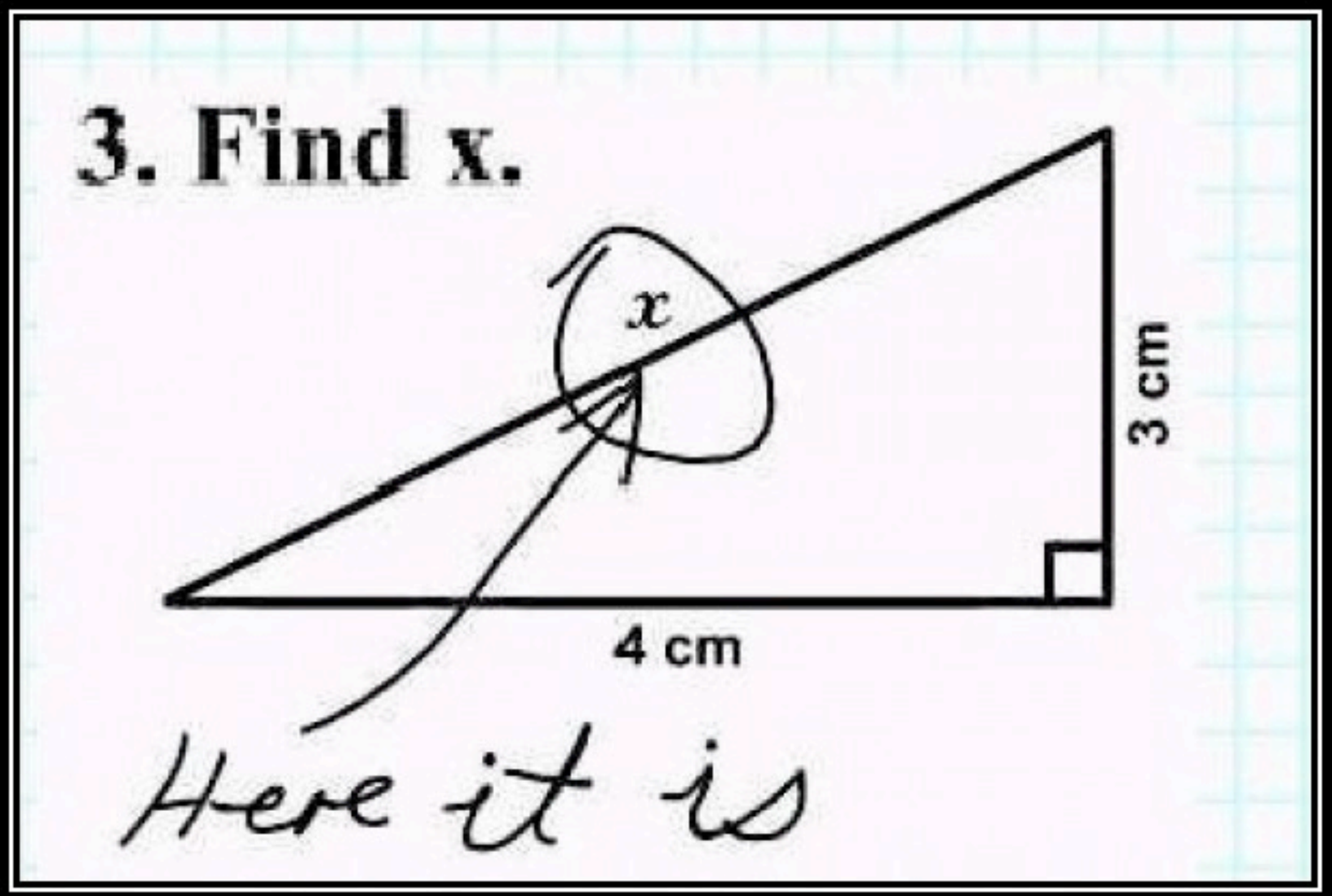

THE #1 PROGRAMMER EXCUSE FOR LEGITIMATELY SLACKING OFF:
"MY CODE'S COMPILING."
HEY! GET BACK TO WORK!
COMPILING!
OH. CARRY ON.


Apache
Con
North America
2011



Apache Con North America 2011

What developers do today to specify target environments is NOT enough



SIMPLICITY

The simplest solutions are often the cleverest
They are also usually wrong

Ops

requirements

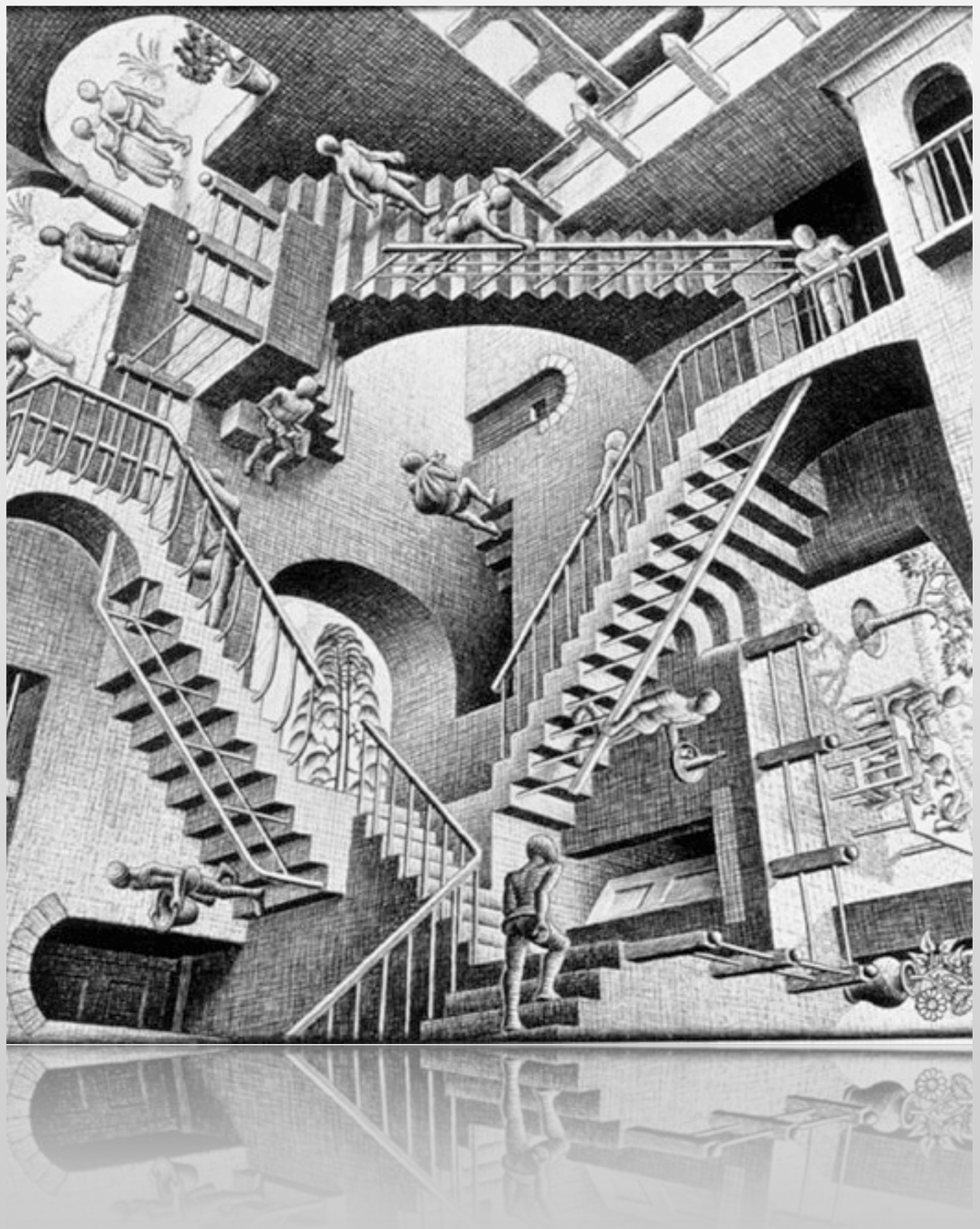
Operating System

config files

packages installed

multi stage configurations

- dev
- QA
- pre-production
- production



Deployment

How do I deploy this?

documentation

manual steps

prone to errors

Cloud

How do I deploy this?

to hundreds of servers

Infrastructure as Code

Follow development best practices

tagging

branching

releasing

dev, QA, production

DevOps

@DEVOPS_BORAT

DevOps Borat

To make error is human. To propagate error to all server in automatic way is #devops.

#devops.

Should I worry about my OPS job?

yes

my job is to make other people's jobs unnecessary

~~yes~~ no

you should see the NOOPS guys

DevOps is NOT about the tools

Nice, BUT

how can I implement **IT**

Tools can enable change in behavior and eventually change culture

Patrick Debois

ApacheCon North America 2011

Man is the only animal that changes environment instead of adapting

(using tools)

DevOps tools

everyone thinks is intelligent enough

every tool thinks it's cloud enabled

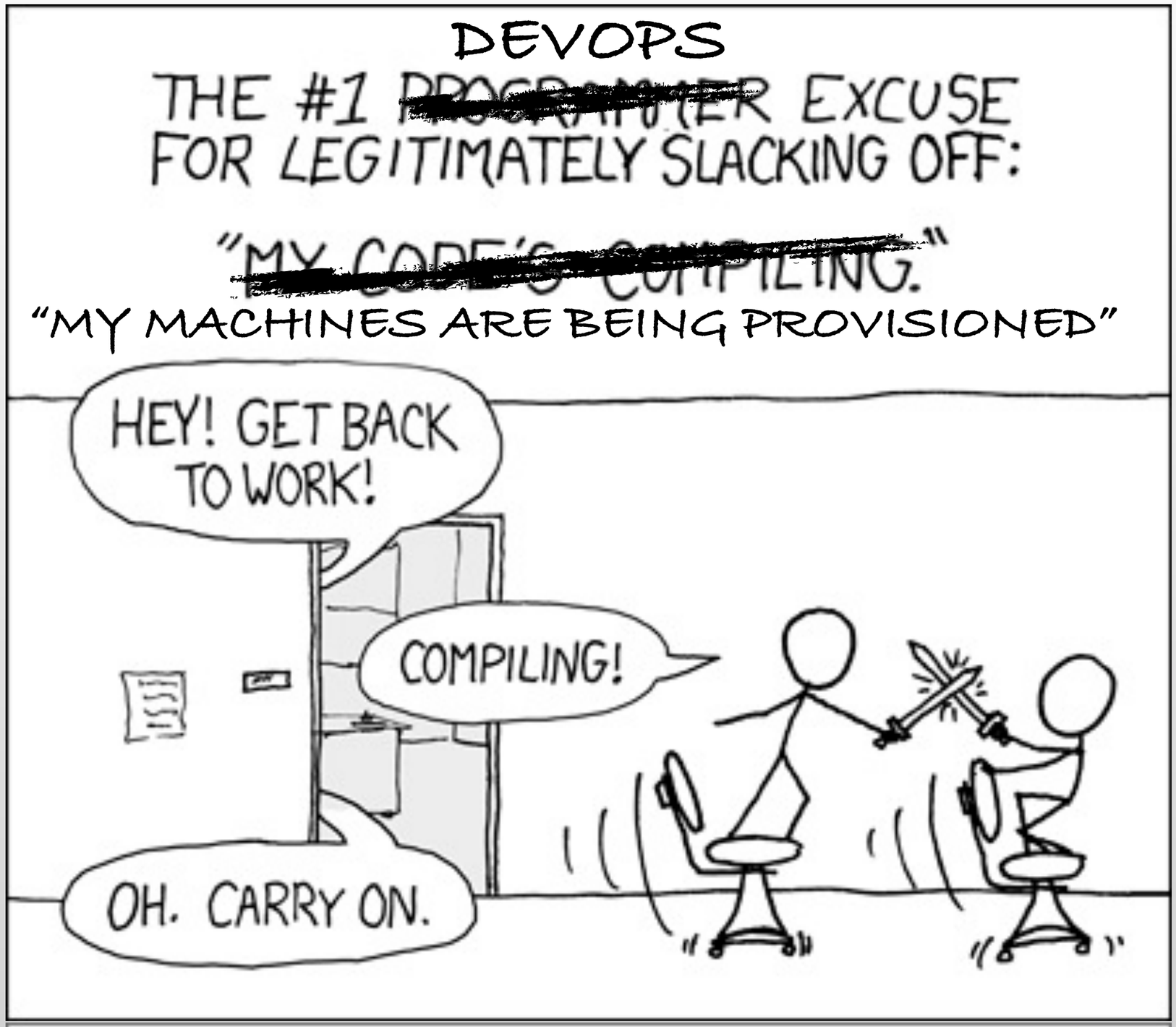
every tool thinks it's DevOps

DevOps tools: infrastructure automation

infrastructure as code

it's all invented, now it's standardized

Chef


DEVOPS
THE #1 [redacted] EXCUSE
FOR LEGITIMATELY SLACKING OFF:
"[redacted]."
"MY MACHINES ARE BEING PROVISIONED"
HEY! GET BACK TO WORK!
COMPILING!
OH. CARRY ON.

Apache Con North America 2011

Puppet

infrastructure IS code

declarative model

state vs process - no scripting

manifests

ruby

ERB templates

master - agent architecture

Puppet file structure

/etc/puppet

manifests/site.pp

manifests/nodes.pp

Puppet example

```
user { 'dave':
  ensure     => present,
  uid        => '507',
  gid        => 'admin',
  shell      => '/bin/zsh',
  home       => '/home/dave',
  managehome => true,
}
file {'/tmp/test1':
  ensure  => present,
  content => "Hi.",
}
```

Vagrant

Vagrant

Oracle VirtualBox cmdline automation

Easy Puppet and Chef provisioning

Keep VM configuration for different projects

Share boxes and configuration files across teams

base box + configuration files

Vagrant base boxes

www.vagrantbox.es

anywhere! just (big) files

using Vagrant

```
$ gem install vagrant
$ vagrant box add centos-6.0-x86_64 \
    http://dl.dropbox.com/u/1627760/centos-6.0-x86_64.box
$ vagrant init myproject
$ vagrant up
$ vagrant ssh
$ vagrant suspend
$ vagrant resume
$ vagrant destroy
```

Vagrant

```
Vagrant::Config.run do |config|

  # Every Vagrant virtual environment requires a box to build off of.
  config.vm.box = "centos-6.0-x86_64"

  # The url from where the 'config.vm.box' box will be fetched
  config.vm.box_url = "http://dl.dropbox.com/u/1627760/centos-6.0-x86_64.box"

  # Boot with a GUI so you can see the screen. (Default is headless)
  #config.vm.boot_mode = :gui

  # Assign this VM to a host only network IP, allowing you to access it via the IP.
  # config.vm.network "33.33.33.10"

  # Forward a port from the guest to the host, which allows for outside
  # computers to access the VM, whereas host only networking does not.
  config.vm.forward_port "sonar", 9000, 19000

  # Enable provisioning with Puppet stand alone.
  config.vm.share_folder("templates", "/tmp/vagrant-puppet/templates", "templates")

  config.vm.provision :puppet do |puppet|
    puppet.manifest_file = "base.pp"
    puppet.module_path = "mymodules"
    puppet.options = ["--templatedir","/tmp/vagrant-puppet/templates"]
    puppet.options = "-v -d"
  end
end
```

manifests/base.pp

```
package { jdk:
   ensure => installed,
   name   => $operatingsystem ? {
      centOS => "java-1.6.0-openjdk-devel",
      Ubuntu => "openjdk-6-jdk",
      default => "jdk",
   },
}
```

VeeWee

VeeWee

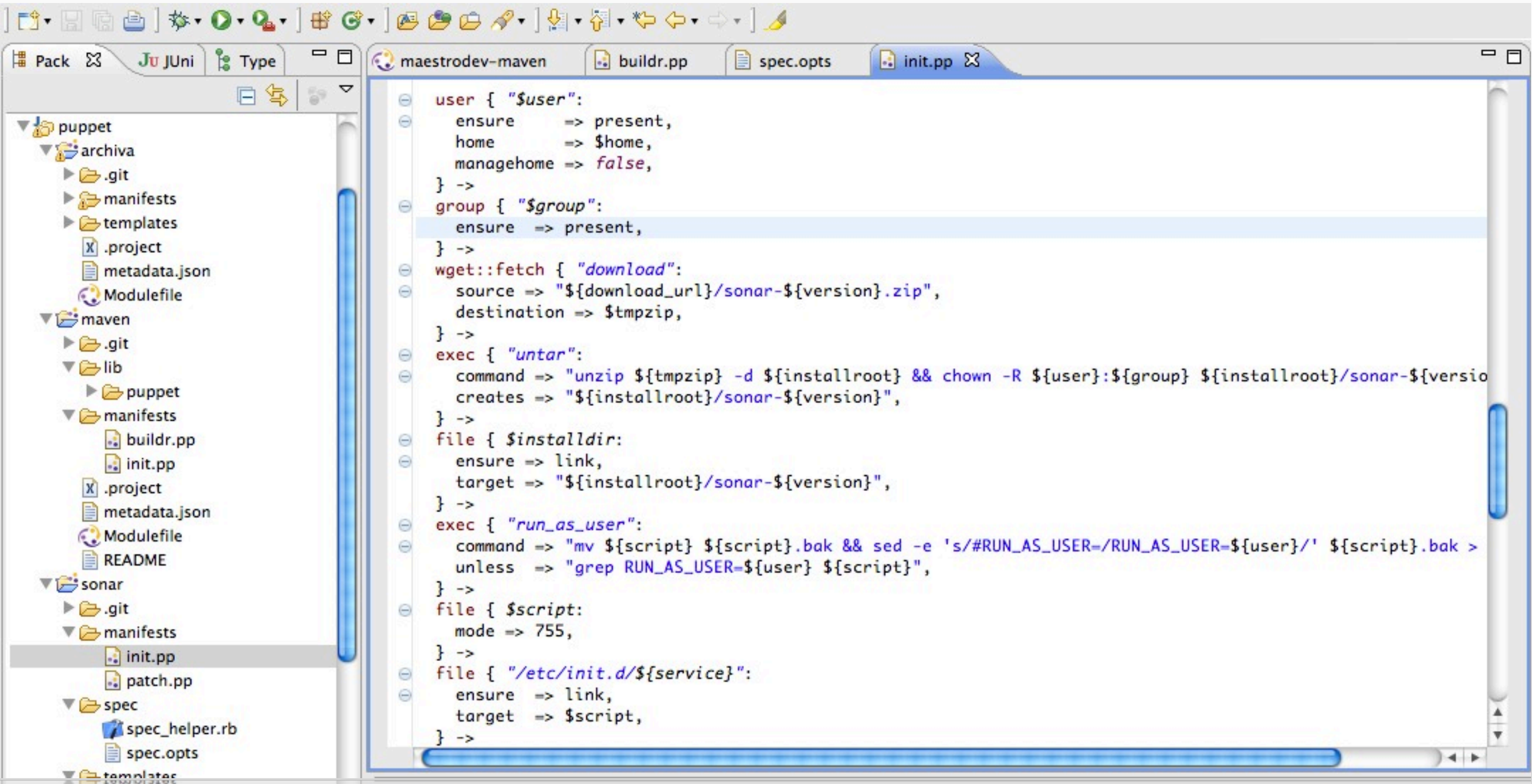
easily build Vagrant base boxes

https://github.com/jedi4ever/veewee

using VeeWee

```
$ gem install veewee
$ vagrant basebox templates
$ vagrant basebox define 'my-ubuntu-server'
    'ubuntu-11.04-server-amd64'
# customize definitions/my-ubuntu-server
$ vagrant basebox build 'my-ubuntu-server'
$ vagrant basebox validate 'my-ubuntu-server'
$ vagrant basebox export 'my-ubuntu-server'
$ vagrant box add 'my-ubuntu-server'
    'my-ubuntu-server.box'
$ vagrant init 'my-ubuntu-server'
```

Geppetto



http://cloudsmith.github.com/geppetto

Apache Con North America 2011

Puppet DSL

Puppet resources

```
user { 'dave':
    ensure      => present,
    uid         => '507',
    gid         => 'admin',
    shell       => '/bin/zsh',
    home        => '/home/dave',
    managehome  => true,
}
```

Puppet resources

```
$ puppet resource user root

user { 'root':
    home => '/var/root',
    shell => '/bin/sh',
    uid => '0',
    ensure => 'present',
    password => '*',
    gid => '0',
    comment => 'System Administrator'
}
```

Puppet resources

```
$ puppet resource user dave
ensure=present shell="/bin/zsh" home="/
home/dave" managehome=true
```

```
notice: /User[dave]/ensure: created

user { 'dave':
    ensure => 'present',
    home => '/home/dave',
    shell => '/bin/zsh'
}
```

Puppet resources

```
file {'testfile':
  path    => '/tmp/testfile',
  ensure  => present,
  mode    => 0640,
  content => "I'm a test file.",
}
notify {"I'm notifying you.":}
```

Puppet standalone

```
$ puppet apply my_test_manifest.pp
```

ordering

```
file {'/tmp/test1':
  ensure  => present,
  content => "Hi.",
}

notify {'/tmp/test1 has already
been synced.':
  require => File['/tmp/test1'],
}
```

ordering

```
file {'/tmp/test1':
  ensure  => present,
  content => "Hi.",
} ->

notify {'/tmp/test1 has already
been synced.':}
```

package / file /service / subscribe

```
package { 'openssh-server':
  ensure => present,
  before => File['/etc/ssh/sshd_config'],
}

file { '/etc/ssh/sshd_config':
  ensure => file,
  mode   => 600,
  source => '/root/learning-manifests/sshd_config',
}

service { 'sshd':
  ensure     => running,
  enable     => true,
  hasrestart => true,
  hasstatus  => true,
  subscribe  => File['/etc/ssh/sshd_config'],
}
```


variables

```
$longthing = "Imagine I have something really
long in here. Like an SSH key, let's say."

file {'authorized_keys':
  path    => '/root/.ssh/authorized_keys',
  content => $longthing,
}
```

facts

```
host {'self':
  ensure        => present,
  name          => $::fqdn,
  host_aliases => ['puppet', $::hostname],
  ip            => $::ipaddress,
}

file {'motd':
  ensure  => file,
  path    => '/etc/motd',
  mode    => 0644,
  content => "Welcome to ${::hostname},\na $
{::operatingsystem} island in the sea of ${::domain}.
\n",
}
```

conditionals

```
if $is_virtual {
  service {'ntpd':
    ensure => stopped,
    enable => false,
  }
}
else {
  service { 'ntpd':
    name       => 'ntpd',
    ensure     => running,
    enable     => true,
    hasrestart => true,
    require => Package['ntp'],
  }
}
```

conditionals

```
case $operatingsystem {
   centos, redhat: { $apache = "httpd" }
   debian, ubuntu: { $apache = "apache2" }
   default: { fail("Unrecognized operating system for
webserver") }
}

$apache = $operatingsystem ? {
     centos                    => 'httpd',
     redhat                    => 'httpd',
     /(?i)(ubuntu|debian)/ => "apache2-$1",
       # (Don't actually use that package name.)
     default                   => undef,
   }
```

class definition

```
class ntp {

  package { 'ntp':
    ensure => installed,
  }

  service { 'ntp':
    name      => 'ntpd',
    ensure    => running,
    enable    => true,
    subscribe => File['ntp.conf'],
  }
}
```

class declaration (1)

```
class {'ntp': }
```

class declaration (2)

```
include ntp
```

parameterized classes

```
class paramclassexample ($value1, $value2 = "Default
value") {
  notify {"Value 1 is ${value1}.":}
  notify {"Value 2 is ${value2}.":}
}
```

```
class {'paramclassexample':
  value1 => 'Something',
  value2 => 'Something else',
}

class {'paramclassexample':
  value1 => 'Something',
}
```

modules

```
{module}/

    files/
    lib/
    manifests/
        init.pp
        {class}.pp
        {defined type}.pp
        {namespace}/
            {class}.pp
            {class}.pp
    templates/
    tests/
```

templating

```
file {'/etc/foo.conf':
    ensure  => file,
    require => Package['foo'],
    content => template('foo/foo.conf.erb'),
}
```

templates/foo.conf.erb

```
OS is <%= $::operatingsystem %>
```

node configuration

```
# nodes.pp

node 'someserver.domain.com' inherits basenode {
    $web_fqdn = 'www.domain.com'
    include genericwebserver
    include some_other_service
}

node 'ldapmaster.domain.com' inherits basenode {
    include s_ldap::master
}

node 'humanresources.domain.com' inherits basenode {
    include c_humanresources
}
```

Maven and Puppet

What am I doing to automate deployment

Ant tasks plugin

ssh commands

Assembly plugin

Cargo

What can I do to automate deployment

Handle full deployment including infrastructure

not just webapp deployment

Help Ops with clear, automated manifests

Ability to reproduce production environments

in local box using Vagrant / VirtualBox / VMWare

Use the right tool for the right job

Maven-Puppet module

A Maven Puppet module

https://github.com/maestrodev/puppet-maven

fetches Maven artifacts from the repo

manages them with Puppet

no more extra packaging

Requirements

Java JDK

Maven

Installing Maven

```
$repo1 = {
  id => "myrepo",
  username => "myuser",
  password => "mypassword",
  url => "http://repo.acme.com",
}

# Install Maven
class { "maven::maven":
  version => "2.2.1",
} ->

# Create a settings.xml with the repo credentials
class { "maven::settings" :
  servers => [$repo1],
}
```

New Maven type

```
maven { "/tmp/maven-core-2.2.1.jar":
    id => "org.apache.maven:maven-core:jar:2.2.1",
    repos => ["http://repo1.maven.apache.org/maven2",
              "http://mirrors.ibiblio.org/pub/mirrors/maven2"],
}
```


New Maven type

```
maven { "/tmp/maven-core-2.2.1.jar":
   groupId => "org.apache.maven",
   artifactId => "maven-core",
   version => "2.2.1",
   packaging => "jar",
   repos => ["http://repo1.maven.apache.org/maven2",
             "http://mirrors.ibiblio.org/pub/mirrors/maven2"],
}
```


Example code

Available at

http://github.carlossanchez.eu

http://blog.carlossanchez.eu

https://github.com/maestrodev/puppet-modules

MAESTRO DEV

Apache CON North America 2011

Thanks!

http://maestrodev.com
http://carlossanchez.eu

csanchez@maestrodev.com
carlos@apache.org

@csanchez

Photo Credits

Son of Man Lego - Alex Eylar

http://www.flickr.com/photos/hoyvinmayvin/4702772452/

Brick wall - Luis Argerich

http://www.flickr.com/photos/lrargerich/4353397797/

Agile vs. Iterative flow - Christopher Little

http://en.wikipedia.org/wiki/File:Agile-vs-iterative-flow.jpg

DevOps - Rajiv.Pant

http://en.wikipedia.org/wiki/File:Devops.png

Pimientos de Padron - Howard Walfish

http://www.flickr.com/photos/h-bomb/4868400647/

Compiling - XKCD

http://xkcd.com/303/

Printer in 1568 - Meggs, Philip B

http://en.wikipedia.org/wiki/File:Printer_in_1568-ce.png

Relativity - M. C. Escher

http://en.wikipedia.org/wiki/File:Escher%27s_Relativity.jpg