Are you aware?

*Aware of **your liability** as a software engineer **!***

About Me

- Marco Pas
- Happy Coder/Software Architect/ DevOps Engineer
- Prototype / First of a kind development
- Doing fun and interesting stuff
- @ Philips Research

Research Areas

- Consumer products
- HealthCare

Some innovations

- Medical X-ray tube
- Mixed Tapes / CD / DVD
- Ambilight TV
- Airfloss

Lets start

with an essential question!

What are we?

Professionals

Our profession

- Developers
- Product Managers
- Agilists
- Testers
- Architects
- Managers
- ...

We are all proud of the things we

create

We as software engineers are awesome

...

We have fun

```
Exception up = new Exception("Something is really wrong.");
throw up;
```

Sometime we are ignorant

```
catch (Exception e) {
 //who cares?
}
```

Sometimes we make mistakes

```
int getRandomize(int randMax)
{
  srand ( time(NULL) );
  int randNum; = rand() % randMax + 1;
  return 2; /* :) */
}
```

We also like to write stories

```
// When I wrote this, only God and I understood what I was doing
// Now, God only knows

----------------------------------------------------------------

// Happy debugging suckers

----------------------------------------------------------------

// Drunk, fix later

----------------------------------------------------------------

// This code sucks, you know it and I know it.
```

But are you also aware of the

consequences

Software is everywhere
and is a challenge/problem

Lets look at some recent challenges ->

Stalled motor

Lack of oxygen

Philips Respironics

Delayed or overdosed medicine

Product recall

CareFusion

Software bug assists

Bank Heist $81 million

Bangladesh Bank

Killing 8500 Patients

"On Paper"

St. Mary's Mercy Medical Center

Releasing 3200 prisoners

too early

Michigan Dept. of Corrections

Network going down

30 million users affected

02

Revealing affairs

Pushing notifications

Uber

Missile strike

State wide alarm

Hawaii

Car emission

if (test) then lower emission

Volkswagen

Money laundering possible

after 10 warnings

ING Banking

How does this effect us?

Law/regulation is coming!

We are are becoming

liable

for the work that we do!

Liability in negligence

Duty of care

- **Detailed testing** of the software before commercial release
- Appropriate use of **automated testing and code quality tools**
- **Notifying customers who have been potentially affected** by a defect in the software

The Programmer's Oath[1]

We need to *regulate ourselves* or others will

- I will, **not produce harmful code.**
- I will, **not knowingly allow code that is defective either in behavior or structure to accumulate**.
- I will, **fearlessly and relentlessly improve my creations at every opportunity.**

---

[1] The Programmer's Oath - https://blog.cleancoder.com/uncle-bob/2015/11/18/TheProgrammersOath.html

Hygiene

How to deploy and enforce hygiene?

Compliance & Security Testing

Validate, weather the system developed meets the organization's prescribed standards or not.

Automate!

Hygiene levels

Application Code

Used Libraries / Dependencies

Containers

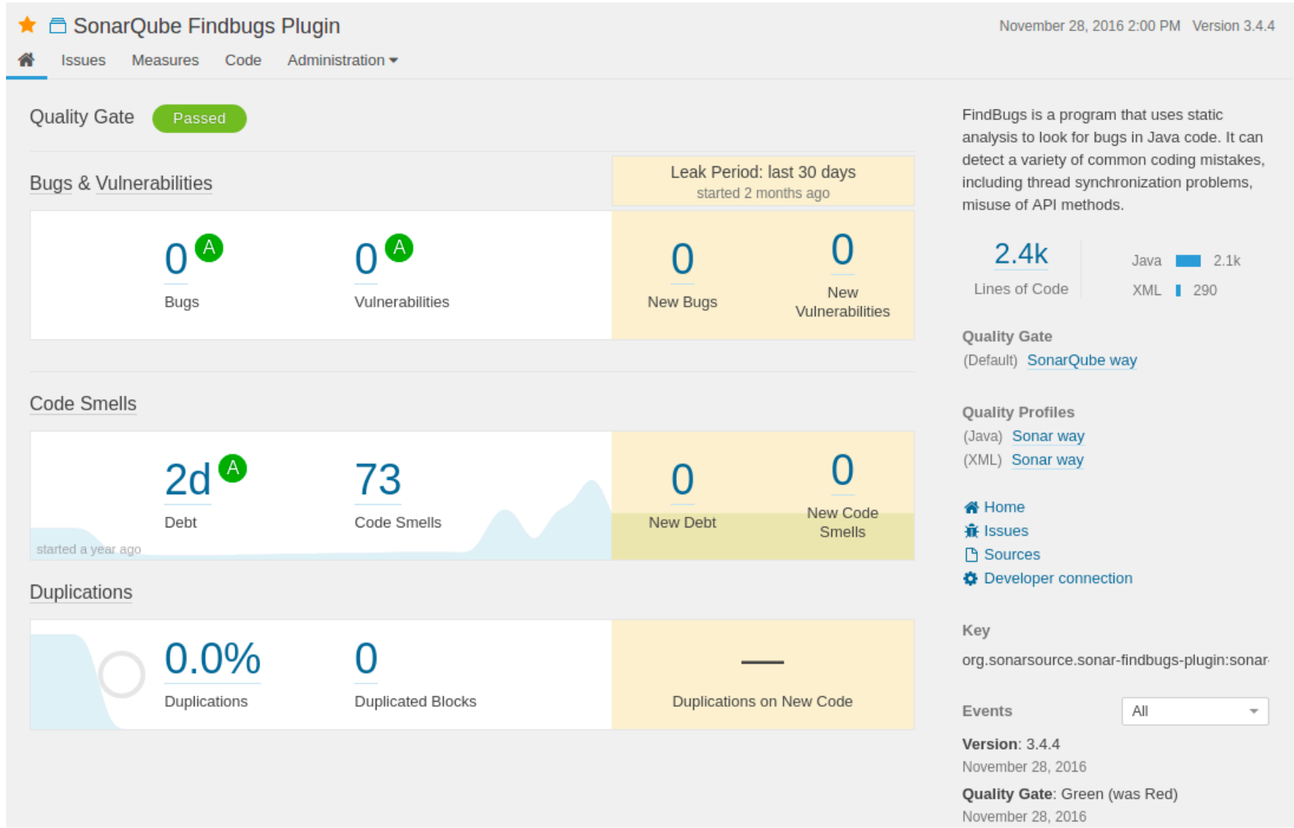
Deployment

Application Code

- Coding Standards
- Quality Attributes
  - Bugs, Code Smells, Coverage, Duplication
- Security Issues
- Predictive Analytics and social patterns

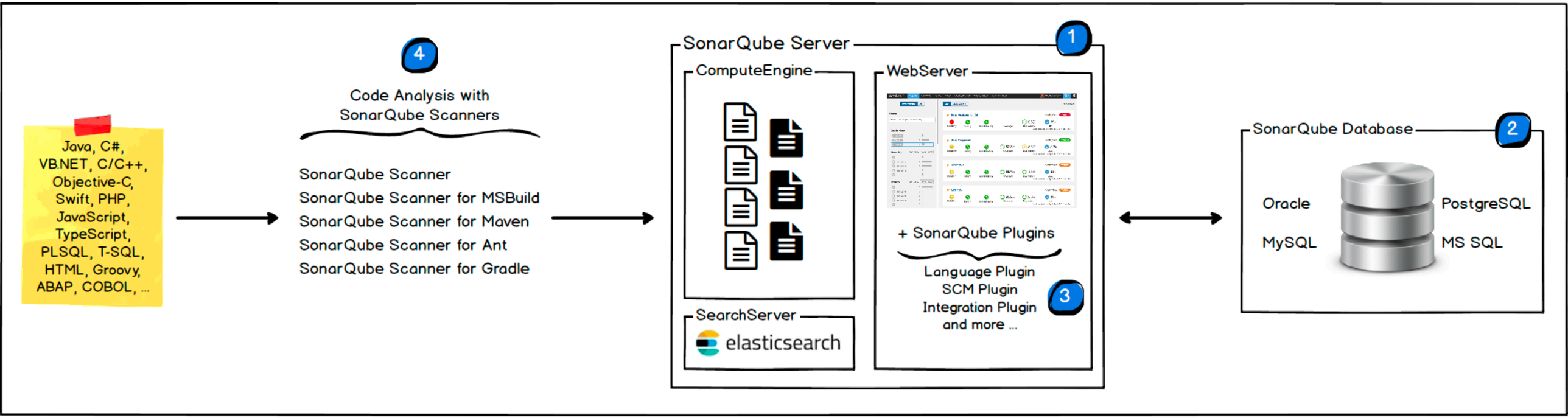
SonarQube[2]

- Continuous Inspection
- Issue detection
- Multi-language
- Centralized:
  - Coding Standards
  - Quality Attributes



[2] SonarQube - https://www.sonarqube.org/

SonarQube Overview



Bad Ex.


E
Reliability Rating on New Code
is worse than A
E
Security Rating on New Code
is worse than A
0.0%
Coverage on New Code
is less than 80.0%
27.7%
Duplicated Lines on New Code
is greater than 3.0%
Bugs
Vulnerabilities
New code: since previous version
started 2 days ago
2.5k E
Bugs
233 E
Vulnerabilities
2.5k E
New Bugs
227 E
New Vulnerabilities
Code Smells
127d A
Debt
started 2 days ago
7.6k
Code Smells
124d A
New Debt
7.4k
New Code Smells
Coverage
0.0%
Coverage
193
Unit Tests
0.0%
Coverage on
46k New Lines to Cover
Duplications
27.4%
Duplications
2k
Duplicated Blocks
27.7%
Duplications on
296k New Lines

Pull Request Feedback

SonarQube @sonarqube commented on commit 18514d4d

Developer

SonarQube analysis reported 8 issues

- 🚫 1 critical
- 🔽 7 minor

Note: The following issues were found on lines that were not modified in the commit. Because these issues can't be reported as line comments, they are summarized here:

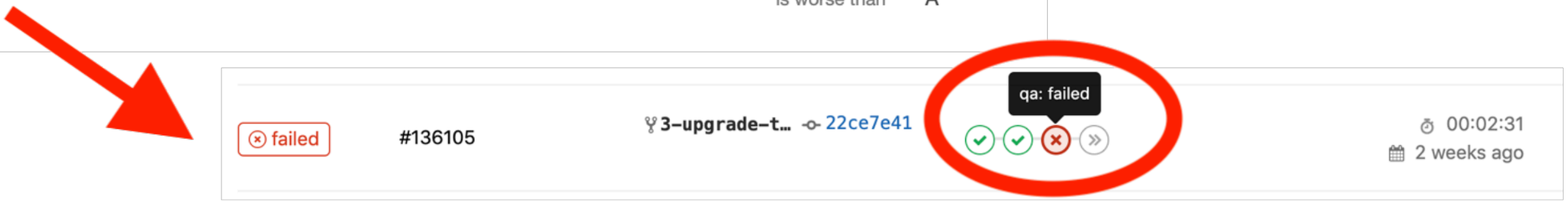
1. 🚫 Replace this persistent entity with a simple POJO or DTO object. 📘
2. 🔽 Make this final field static too. 📘
3. 🔽 Remove this use of "NotEmpty"; it is deprecated. 📘
4. 🔽 Remove this use of "NotEmpty"; it is deprecated. 📘
5. 🔽 Replace "@RequestMapping(method = RequestMethod.GET)" with "@GetMapping" 📘
6. 🔽 Replace "@RequestMapping(method = RequestMethod.GET)" with "@GetMapping" 📘
7. 🔽 Replace "@RequestMapping(method = RequestMethod.POST)" with "@PostMapping" 📘
8. 🔽 Replace "@RequestMapping(method = RequestMethod.POST)" with "@PostMapping" 📘

Quality Gates

Conditions

Only project measures are checked against thresholds. Directories and files are ignored.

Metric	Operator	Error
Coverage on New Code	is less than	80.0%
Duplicated Lines on New Code	is greater than	3.0%
Maintainability Rating on New Code	is worse than	A
Reliability Rating on New Code	is worse than	A
Security Rating on New Code	is worse than	A



Define your own

quality gates

to ensure compliance

Include SonarQube + Test Coverage

```
// file: build.gradle
plugins {
    ...
    id "org.sonarqube" version "2.7.1"
    id "jacoco"
}

sonarqube {
    properties {
        property "sonar.coverage.jacoco.xmlReportPaths", "$buildDir/reports/jacoco/test/jacocoTestReport.xml"
        property "sonar.coverage.exclusions", ["**/Application.java"]
    }
}

jacocoTestReport {
    reports {
        xml.enabled true
    }
}
```

Demo

SonarQube

Application Code

- ✅ Coding Standards
- ✅ Quality Attributes
  - Bugs, Code Smells, Coverage, Duplication
- ✅ Security Issues
- Predictive Analytics and social patterns

Predictive Analysis [3]

[3] CodeScene - https://codescene.io/

The Pragmatic Programmers

Investigator:
Date:
Case #:
Location:

Your Code As a Crime Scene

Use Forensic Techniques
to Arrest Defects, Bottlenecks, and
Bad Design in Your Programs

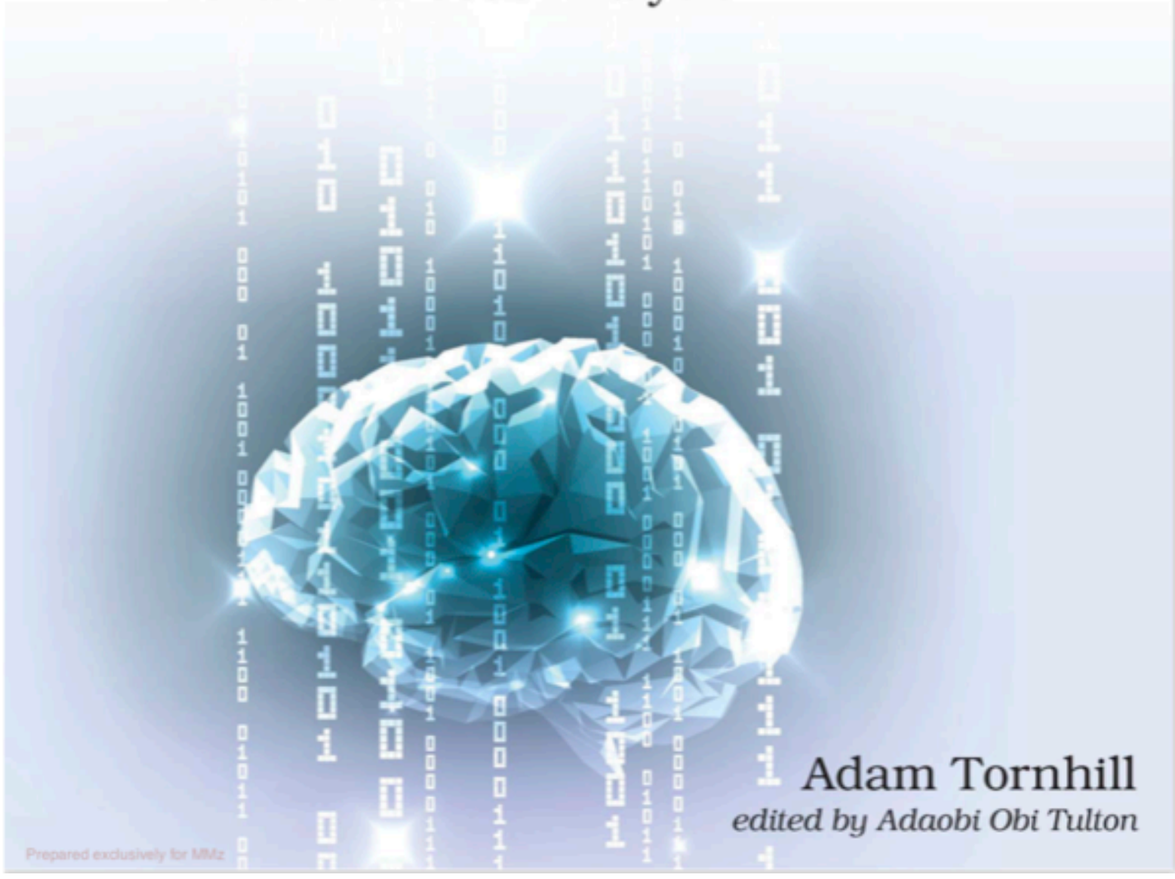
Adam Tornhill
edited by Fahmida Y. Rashid

Foreword by Michael Feathers,
author of Working Effectively
with Legacy Code

The Pragmatic Programmers

Software Design X-Rays

Fix Technical Debt with Behavioral Code Analysis



Adam Tornhill
edited by Adaobi Obi Tulton

Every commit leaves a trace

- Which part of the code might become bottlenecks?
- Which parts of the code will be hard to maintain?
- What is the technical risk when a developer leaves the project?
- Which parts of the code should we improve to get a real productivity and quality gain?
- How is the knowledge distribution between teams in your codebase?

```
hibernate-orm (master) $ git log --pretty=format:'%h   %<(20)%aN   %ai   %s' -25
cc10cbe   Naros                2016-04-04 13:24:22 -0500   HHH-10670 - Removed deprecated ValidTimeAuditStrategy
3152bb0   Martin Simka         2016-01-28 14:48:05 +0100   [HHH-10290] ignore environment property hibernate.jdbc.
18dd88c   Martin Simka         2016-03-04 12:13:31 +0100   [HHH-10587] skip NationalizedIgnoreCaseTest on db2 and
7868596   Jan Martiska         2016-03-07 16:43:15 +0100   HHH-10598 - Oracle JDBC driver can't handle entities wi
e591b70   Martin Simka         2016-03-04 13:22:43 +0100   [HHH-10588] use H2Dialect in ConnectionsReleaseTest
3e95900   Danny02              2016-03-07 20:00:56 +0100   HHH-10612 - Check for support of RefCursor in Java 8
e861182   Martin Simka         2016-03-24 13:35:32 +0100   [HHH-10640] fix wrong expected statement on DB2
cbdab9d   Zhenlei Huang        2016-03-28 03:16:04 +0800   HHH-10649 - When 2LC enabled, flush session and then re
a68a6c6   Martin Simka         2016-03-24 14:22:07 +0100   [HHH-10641] Fix identifier is too long exception for te
7c75a92   Zhenlei Huang        2016-03-29 10:09:03 +0800   HHH-10652 - The HHH-10631 test makes wrong assertion
3d04839   Steve Ebersole       2016-03-31 12:39:08 -0500   HHH-10664 - Prep 6.0 feature branch - target Hibernate
fa0db89   Steve Ebersole       2016-03-31 12:27:52 -0500   HHH-10664 - Prep 6.0 feature branch - merge hibernate-j
8ddd61b   Steve Ebersole       2016-03-31 12:04:10 -0500   HHH-10664 - Prep 6.0 feature branch - baseline Java 8
9570f11   Vlad Mihalcea        2016-03-31 14:23:06 +0300   User Guide grammatical corrections
da5aae7   Vlad Mihalcea        2016-03-31 12:23:08 +0300   Add JPA-related configuration properties in the new Use
985229f   Vlad Mihalcea        2016-03-31 09:09:02 +0300   HHH-10662 - Fix inconsistencies between quoting-related
de6d80a   Dominique Toupin     2016-01-18 09:51:54 -0500   HHH-10456: Report the class name were the invalid Prima
849c4d2   Vlad Mihalcea        2016-03-28 13:35:37 +0300   Document available Hibernate configurations
86b49a0   Sanne Grinovero      2016-03-29 16:39:20 +0100   HHH-10657 Make 'none' a valid option for hibernate.hbm2
68298bb   Andrea Boriero       2016-03-30 10:55:59 +0200   HHH-10650 - Hibernate SchemaExport does not filter Fore
add68bb   Andrea Boriero       2016-03-29 18:04:48 +0200   Add mariadb config to be used with gradle processTestRe
6036f00   yinzara              2016-01-14 08:55:25 -0800   HHH-10429 - Change SimpleValue isIdentityColumn method
a634cce   Martin Simka         2016-03-07 13:32:58 +0100   [HHH-10596] add missing dialect-scope for oracle12c
4c7525d   Andrea Boriero       2016-03-22 17:42:45 +0000   HHH-10632 - Fix comment on column missing seperator bet
feab506   Andrea Boriero       2016-03-22 16:50:03 +0000   HHH-10632 - Add test for issue
```

Does Measuring Code Change Improve Fault Prediction?

Robert M. Bell, Thomas J. Ostrand, Elaine J. Weyuker
AT&T Labs - Research
180 Park Avenue
Florham Park, NJ 07932
(rbell,ostrand,weyuker)@research.att.com

Where the Bugs Are

Thomas J. Ostrand
AT&T Labs - Research
180 Park Avenue
Florham Park, NJ 07932
ostrand@research.att.com

Elaine J. Weyuker
AT&T Labs - Research
180 Park Avenue
Florham Park, NJ 07932
weyuker@research.att.com

Robert M. Bell
AT&T Labs - Research
180 Park Avenue
Florham Park, NJ 07932
rbell@research.att.com

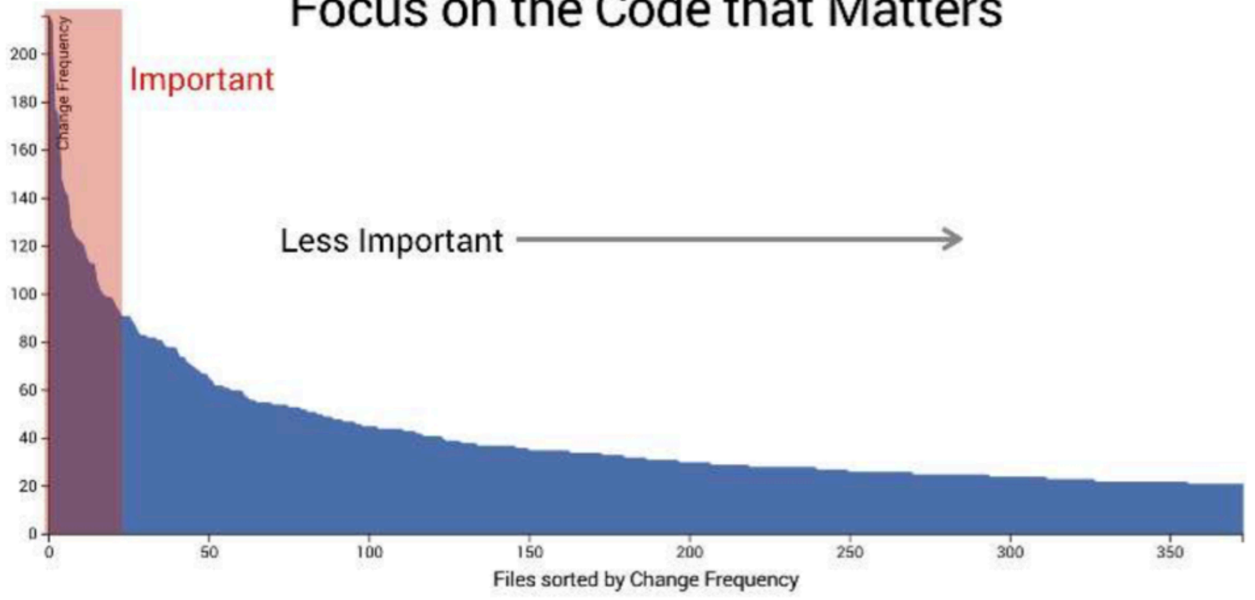
A Comparative Analysis of the Efficiency of Change Metrics and Static Code Attributes for Defect Prediction

Raimund Moser
Free University of Bolzano-Bozen
Piazza Domenicani 3
I-39100 Bolzano, Italy
+39 0471016138
Raimund.Moser@unibz.it

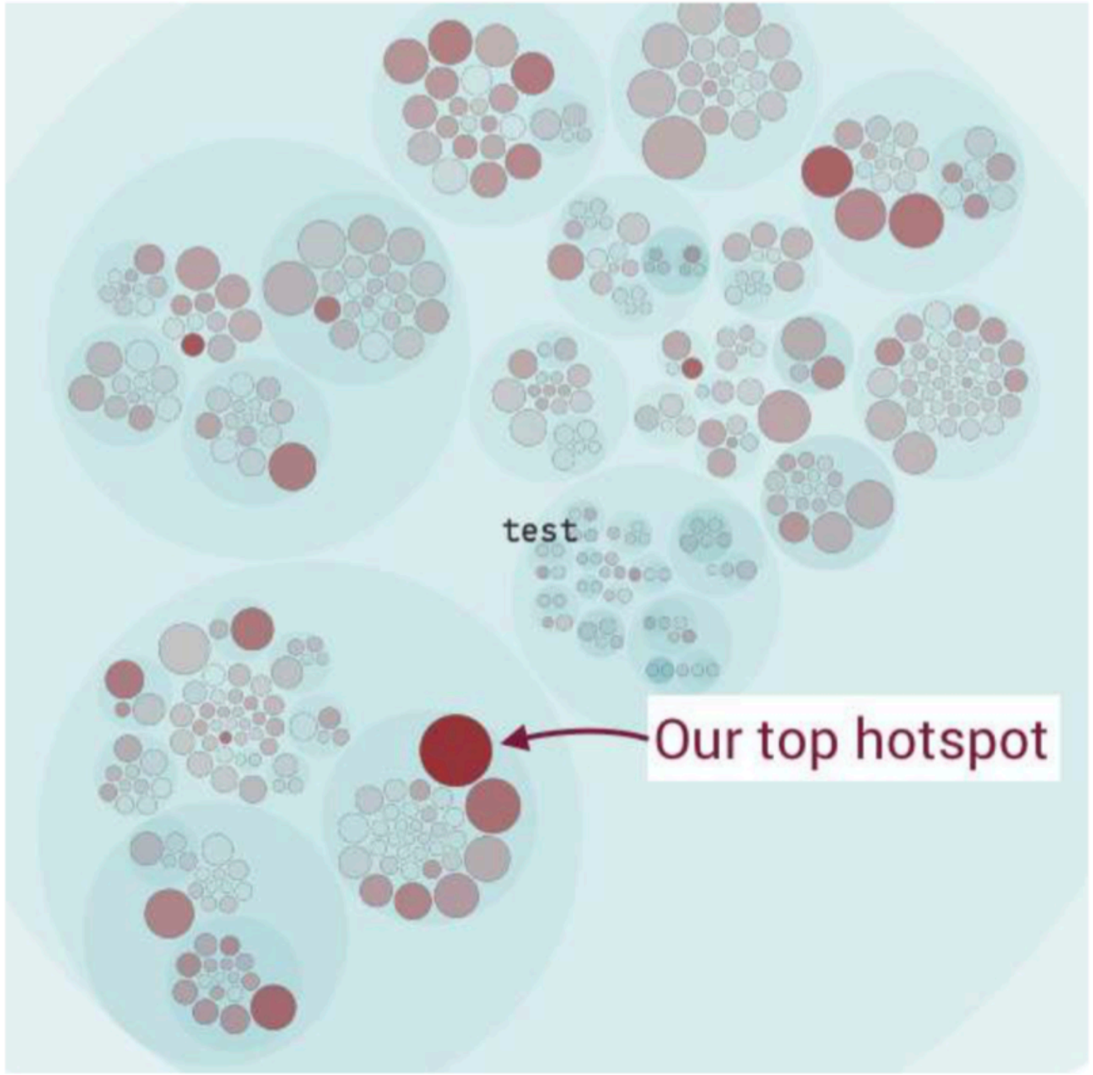
Witold Pedrycz
University of Alberta
T6G 2V4 Edmonton
Alberta, Canada
+1 7804923333
pedrycz@ee.ualberta.ca

Giancarlo Succi
Free University of Bolzano-Bozen
Piazza Domenicani 3
I-39100 Bolzano, Italy
+39 0471016130
Giancarlo.Succi@unibz.it

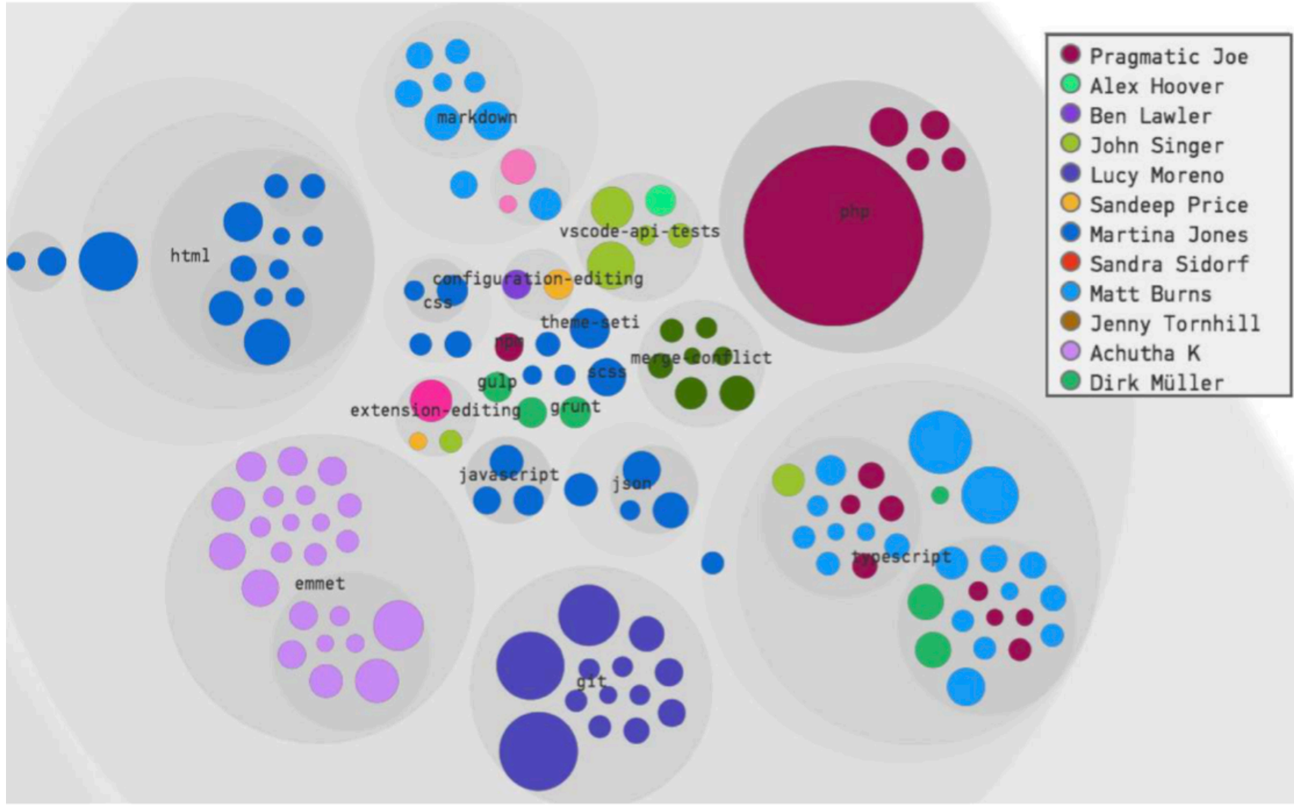
Focus on the Code that Matters



Hotspots



Knowledge distribution



Demo

CodeScene

Application Code

- ✅ Coding Standards
- ✅ Quality Attributes
  - Bugs, Code Smells, Coverage, Duplication
- ✅ Security Issues
- ✅ Predictive Analytics and social patterns

Hygiene levels

Application Code

Used Libraries / Dependencies

Containers

Deployment

Used Libraries / Dependencies

- License Management
- Security Incidents (CVE)

```
import com.github.jk1.license.render.*

plugins {
    ...
    id 'org.owasp.dependencycheck' version '5.0.0-M3.1'
    id 'com.github.jk1.dependency-license-report' version '1.6'
}

licenseReport {
    renderers = [new InventoryHtmlReportRenderer()]
}
```

Demo

License Management / Security Incidents

Used Libraries / Dependencies

- ✅ License Management
- ✅ Security Incidents (CVE)

Hygiene levels

Application Code

Used Libraries / Dependencies

Containers

Deployment

Containers

- Correctly build?
- Trusted and safe?
- Does not expose and strange ports?
- Contains the correct application and settings?

How do we test and verify?

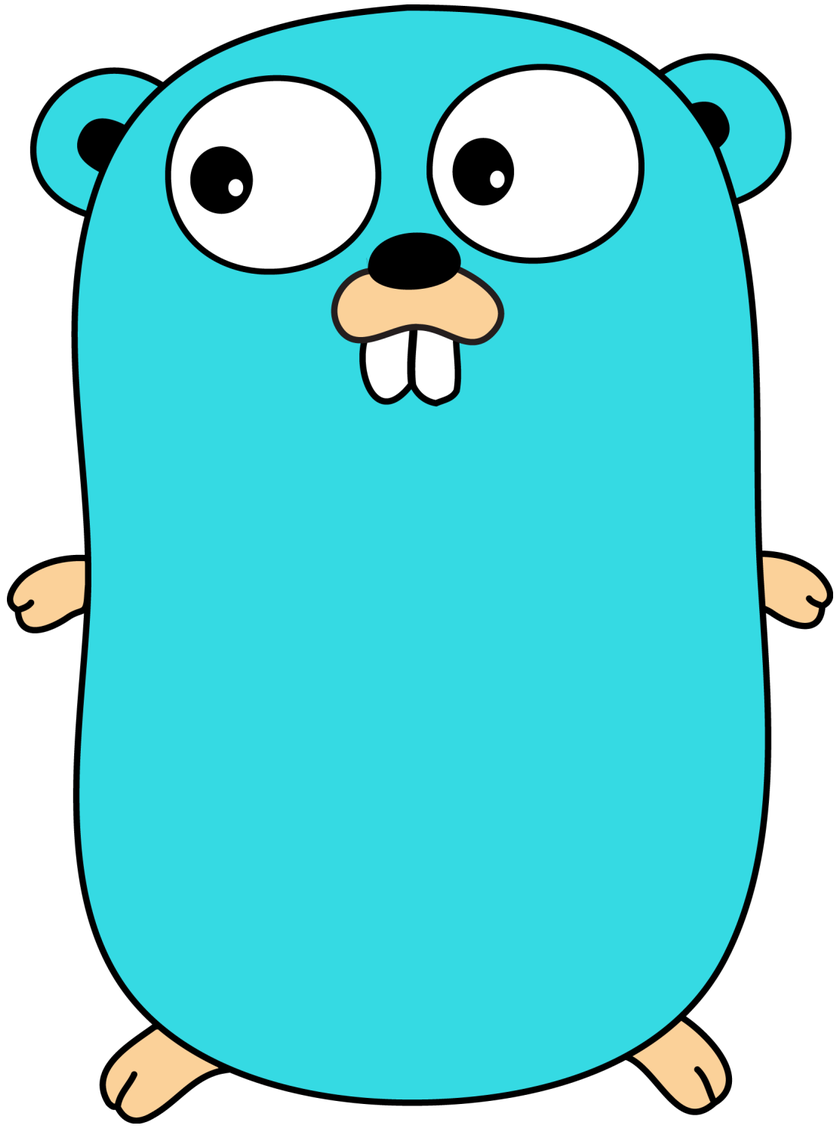
Docker Build File

```
FROM openjdk:11-jdk-slim
RUN mkdir -p /app
COPY *.jar /app/application.jar
WORKDIR /app
ENTRYPOINT ["java","-Dmicronaut.server.port=8020", "-jar","application.jar"]
```

Image ➡ Container

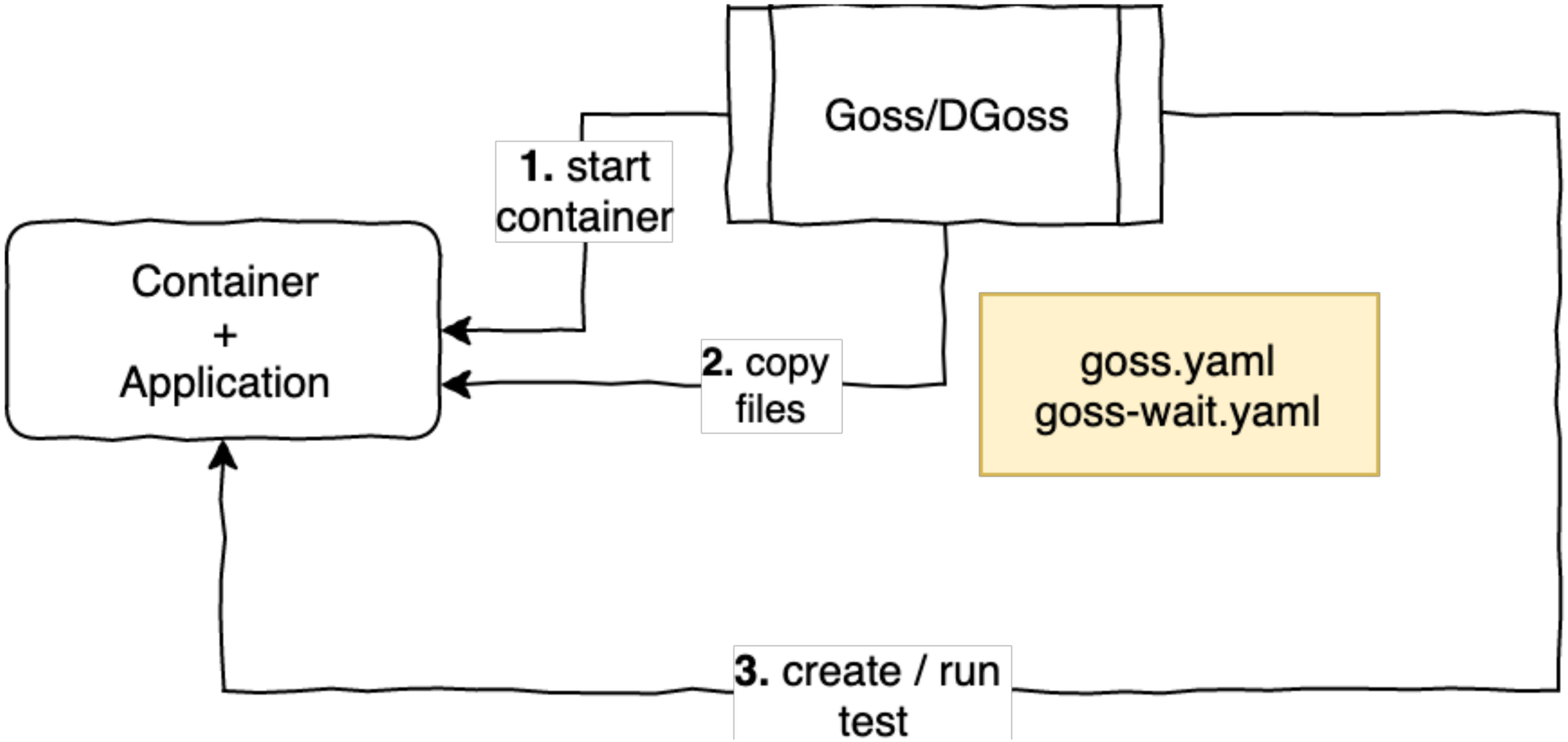
Goss[4]

- YAML based **serverspec** alternative for validating a server's configuration
- Writing tests by allowing the user to **generate tests from the current system state**
- Test suite can be **executed, waited-on, or served as a health endpoint**
- Runs on the target!!



[4] Goss - Quick and Easy server testing/validation - https://github.com/aelsabbahy/goss

Goss Overview



Simple test!

```
// file: goss.yaml

http:
  http://localhost:8020/helloworld/:
    status: 200
```

```yaml
user:
  sshd:
    title: UID must be between 50-100, GID doesn't matter. home is flexible
    meta:
      desc: Ensure sshd is enabled and running since it's needed for system management
      sev: 5
    exists: true
    uid:
      # Validate that UID is between 50 and 100
      and:
        gt: 50
        lt: 100
    home:
      # Home can be any of the following
      or:
      - /var/empty/sshd
      - /var/run/sshd

package:
  kernel:
    installed: true
    versions:
      # Must have 3 kernels and none of them can be 4.4.0
      and:
      - have-len: 3
      - not:
          contain-element: 4.4.0
```

Supported resources

- package - add new package
- file - add new file
- addr - add new remote address:port - ex: google.com:80
- port - add new listening [protocol]:port - ex: 80 or udp:123
- service - add new service
- user - add new user
- group - add new group
- command - add new command
- dns - add new dns
- process - add new process name
- kernel-param - add new kernel-param
- mount - add new mount
- interface - add new network interface
- http - add new network http url
- goss - add new goss file, it will be imported from this one
- matching - test for matches in supplied content

Creating/Running tests

```
// create/edit tests
$ dgoss edit <container-name>
INFO: Run goss add/autoadd to add resources
# goss add http http://localhost:8020/helloworld/
# exit
INFO: Copied '/goss/goss.yaml' from container to '.'
// run test
$ dgoss run <container-name>
```

Creating/Running tests

```
// create/edit tests
$ dgoss edit <container-name>
INFO: Run goss add/autoadd to add resources
# goss add http http://localhost:8020/helloworld/
# exit
INFO: Copied '/goss/goss.yaml' from container to '.'
// run test
$ dgoss run <container-name>
```

Creating/Running tests

```
// create/edit tests
$ dgoss edit <container-name>
INFO: Run goss add/autoadd to add resources
# goss add http http://localhost:8020/helloworld/
# exit
INFO: Copied '/goss/goss.yaml' from container to '.'
// run test
$ dgoss run <container-name>
```

Demo

Goss

Inspec - compliance as code

InSpec is an **open-source** testing framework for infrastructure with a **human-readable language** for specifying **compliance, security and other policy requirements**.

Why Inspec

- It's open source
- Development supported by Chef Software Inc.
- Awesome community (Slack)
- Resourch rich!!
- Can run anywhere (local machine, over ssh, docker, winrm)
- Written in Ruby

Questions and answers

InSpec provides an incredibly easy way to answer questions such as:

- Is package “my*app” installed, “my*service” running?
- Is the SSH server configured to only accept protocol version 2?
- Is the “max*allowed*packet” setting in the “mysql” section of “/etc/my.cnf” set to “16M”?

Demo

Inspec Shell

Simple test(s)

```
describe file('/etc/myapp.conf') do
  it { should exist }
  its ('mode') { should cmp '0644' }
end

describe package('nginx') do
  it { should be_installed }
end

describe port.where { protocol =~ /tcp/ && port > 22 && port < 80 } do
  it { should_not be_listening }
end
```

Inspec Resources

- Operating System
- Amazon WebServices
- Azure
- Google Cloud

Targets

```
# Login to remote machine using ssh as root
$ inspec shell -t ssh://root@192.168.64.2:11022

# Login to hostname on port 1234 as user using given ssh key
$ inspec shell -t ssh://user@hostname:1234 -i /path/to/user_key

# Login to windowsmachine over WinRM as UserName
$ inspec shell -t winrm://UserName:Password@windowsmachine:1234

# Login to a Docker container
$ inspec shell -t docker://container_id
```

When no target then local is assumed!

Docker resource examples

```
describe docker_image('alpine:latest') do
  it { should exist }
  its('id') { should eq 'sha256:4a415e...a526' }
  its('repo') { should eq 'alpine' }
  its('tag') { should eq 'latest' }
end

describe docker.images do
    its('repositories') { should_not include 'insecure_repository' }
end

describe docker.containers do
    its('images') { should_not include 'this-image-should-not-be-used-anymore:latest' }
end
```

Docker resource examples

```
describe docker_image('alpine:latest') do
  it { should exist }
  its('id') { should eq 'sha256:4a415e...a526' }
  its('repo') { should eq 'alpine' }
  its('tag') { should eq 'latest' }
end

describe docker.images do
    its('repositories') { should_not include 'insecure_repository' }
end

describe docker.containers do
    its('images') { should_not include 'this-image-should-not-be-used-anymore:latest' }
end
```

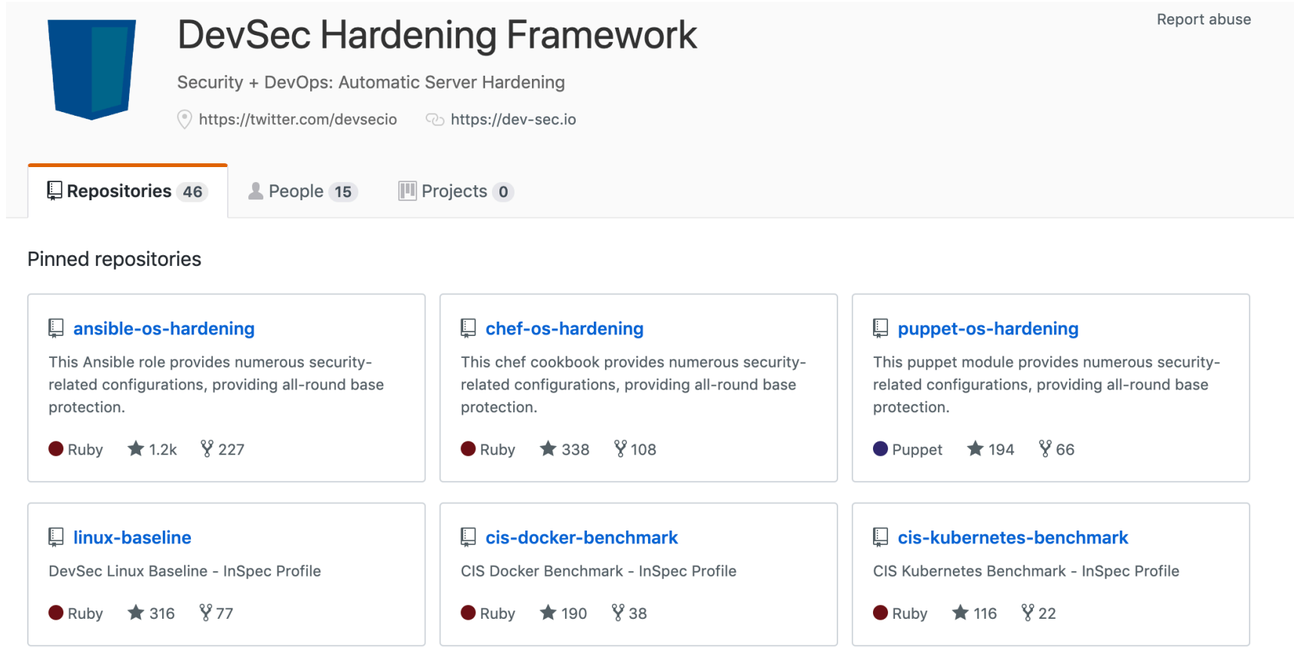
Docker resource examples

```
describe docker_image('alpine:latest') do
  it { should exist }
  its('id') { should eq 'sha256:4a415e...a526' }
  its('repo') { should eq 'alpine' }
  its('tag') { should eq 'latest' }
end

describe docker.images do
    its('repositories') { should_not include 'insecure_repository' }
end

describe docker.containers do
    its('images') { should_not include 'this-image-should-not-be-used-anymore:latest' }
end
```

Inspec Profiles[6]

Describe best configuration practices for specific services

- SSH
- Linux
- Docker
- Postgress
- ...



[6] DevSec.io - https://dev-sec.io

Inspec Controls

```
container_name = attribute('container_name',
                           description: 'Name of the container to be tested.',
                           default: 'Please specify with ATTRIBUTES FLAG --attrs')

control "001-container-should-be-running" do
  impact 1.0
  title "Container should be running."
  desc "The container should be running."

  describe docker_container(container_name) do
    it { should exist }
    it { should be_running }
  end
end
```

Demo

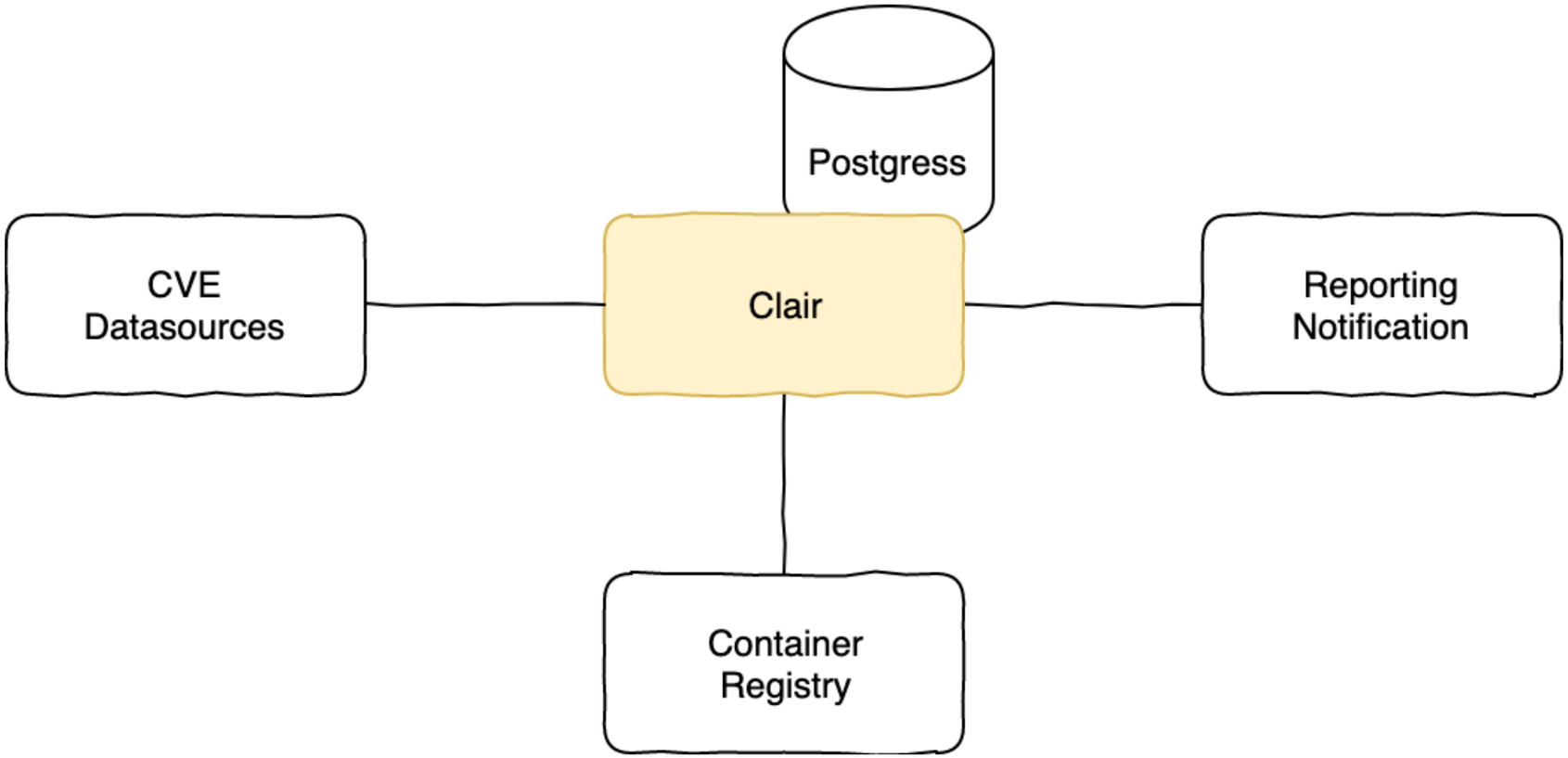
Inspec

Vulnerability Static Analysis for Containers[7]

- Open Source
- Scan layers in images
- Whitelist support
- Easy integration into CI/CD pipelines

[7] Clair Container Scan - https://github.com/coreos/clair

Clair Overview



CVE Data Sources

Data Source	Data Collected	Format	License
Debian Security Bug Tracker	Debian 6, 7, 8, unstable namespaces	dpkg	Debian
Ubuntu CVE Tracker	Ubuntu 12.04, 12.10, 13.04, 14.04, 14.10, 15.04, 15.10, 16.04 namespaces	dpkg	GPLv2
Red Hat Security Data	CentOS 5, 6, 7 namespaces	rpm	CVRF
Oracle Linux Security Data	Oracle Linux 5, 6, 7 namespaces	rpm	CVRF
Amazon Linux Security Advisories	Amazon Linux 2018.03, 2 namespaces	rpm	MIT-0
SUSE OVAL Descriptions	openSUSE, SUSE Linux Enterprise namespaces	rpm	CC-BY-NC-4.0
Alpine SecDB	Alpine 3.3, Alpine 3.4, Alpine 3.5 namespaces	apk	MIT
NIST NVD	Generic Vulnerability Metadata	N/A	Public Domain

Whitelist support

```
generalwhitelist: #Approve CVE for any image
  CVE-2017-6055: XML
  CVE-2017-5586: OpenText
images:
  ubuntu: #Apprive CVE only for ubuntu image, regardles of the version
    CVE-2017-5230: Java
    CVE-2017-5230: XSX
  alpine:
    CVE-2017-3261: SE
```

Demo

Clair

Hygiene Levels

Application Code

Used Libraries / Dependencies

Containers

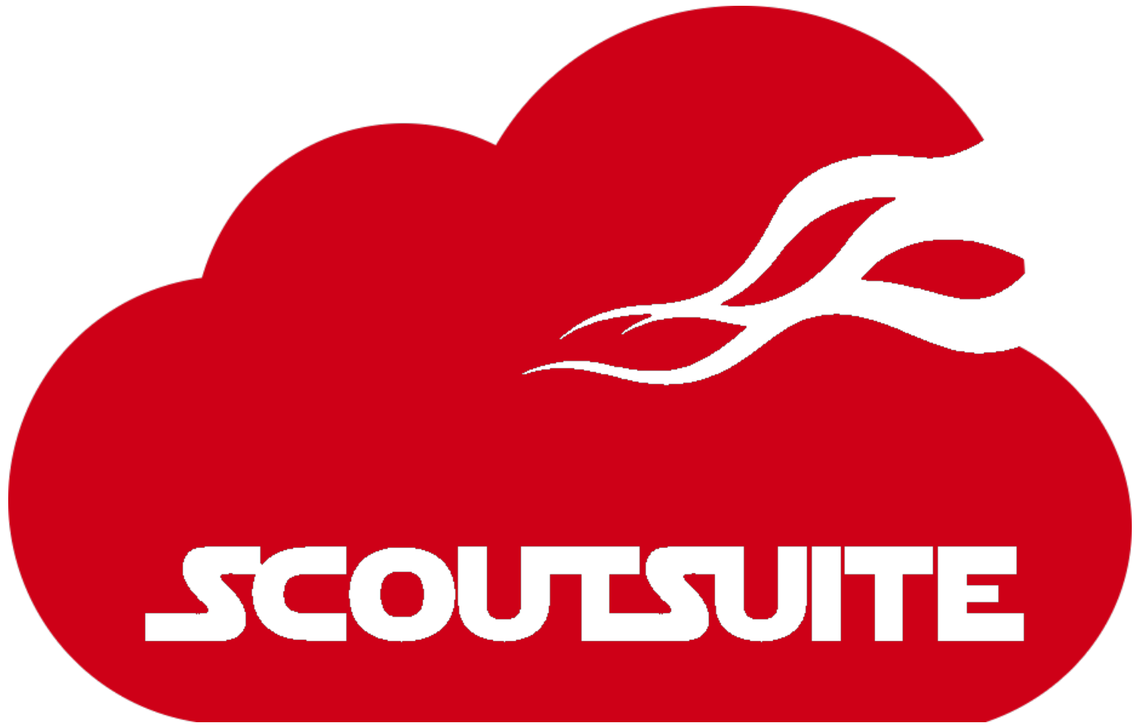
Deployment

Deployment

- Multicloud
- Best practices per vendor
- Auditing

Scout Suite

- Open Source
- Stable and actively maintained
- Multi-cloud
  - Amazon Web Services
  - Microsoft Azure
  - Google Cloud Platform
- Python



Compliance notes

- **does not require AWS/Azure/GCO users**, to complete and submit the AWS Vulnerability / Penetration Testing or contact Microsoft/Google to begin testing

But please read the

Acceptable Use Policy and the Terms of Service

Demo

ScoutSuite

The human factor

Why do quality tools often fail

- **Resistance**
  - There is other important work to do
  - The tools are to slow
  - They dont work correctly
  - Coding standards just suck
  - Management is trying to execute a blame game
- **Nasty corner cases**

Conclusion

Be aware that software is everywhere! You influence peoples lives! So do your utmost best to make them secure and "bug" free. Invest in code analysis, etc..

Take your duty seriously!!

Q & A

Thank you!

All sources for the presentation can be found at:

- https://github.com/mpas/compliance-and-security-testing