

Improving Software Quality Using Component Lifecycle Management with Jenkins



Manfred Moser
Sonatype

<http://www.sonatype.com>



About Manfred



- Author and Presenter
 - Maven: The Complete Reference, The Hudson Book, Repository Management with Nexus
 - AndroidTO, AnDevCon, OSCON,...
- Open Source Contributor
 - ksoap2-android, maven-android-sdk-deployer, android-maven-plugin, roboguice, ...
- Sonatype Trainer for Maven, Nexus...
- @simpligility, <http://simpligility.com>



Component Lifecycle Management for Managers



Analysis, control, and monitoring
of component based software



Component Lifecycle Management Straight Up



- Component
 - assemble vs write code
 - using libraries and frameworks
- Lifecycle
 - take care of the application all the time
 - From source code to production and beyond
- Management
 - have a tool do it for you ;-)



Old School. New School.



Then

[illegible]

Written

Now



Assembled

http://commons.wikimedia.org/wiki/File:Lego_Chicago_City_View_2001.jpg

>80% of a typical modern application is assembled from open source components



Exploding Component Usage

Example Central Repository



A Typical Enterprise

000s of applications built on components

Downloads > 100K components annually

Uses components from 000s of projects

Shares components across teams

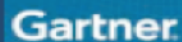
Has limited control over selection and usage



Open Source Is Everywhere



By 2016, OSS will be included in **mission-critical software** portfolios within **99%** of **Global 2000** enterprises, up from **75% in 2010**.



Predicts 2011:
Open Source the Power Behind the Throne
November 2010

But as Jenkins users you know of the power of Open Source already..



The Move To Component-Based Software Development



Things are now possible, that seemed impossible before, **but** we need to

Understand The Risks!

- Security
- Licensing / Intellectual Property / Copyright
- Quality

So they don't negatively impact you...



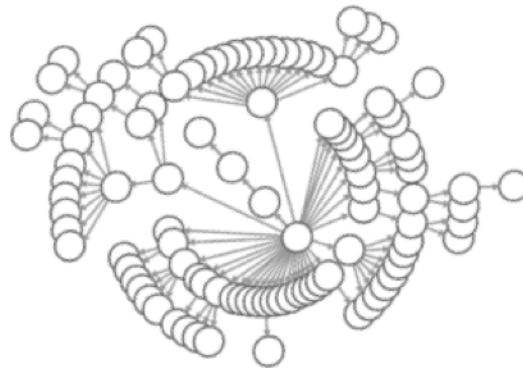
An Immature Ecosystem



No security infrastructure

No Update Alerting

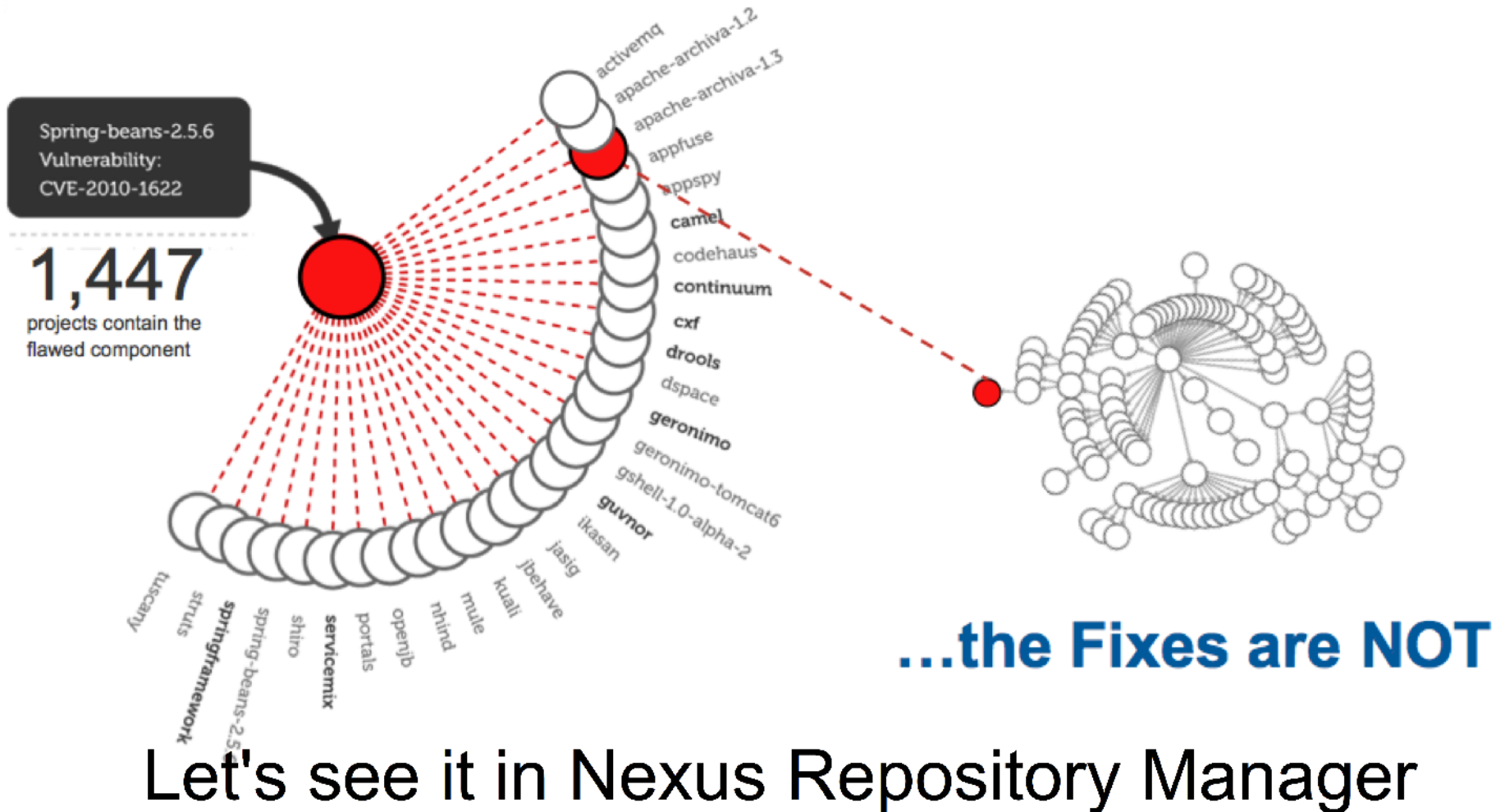
No standard licensing mechanisms



...and complex dependencies amplify the problem



Issues Are Viral...



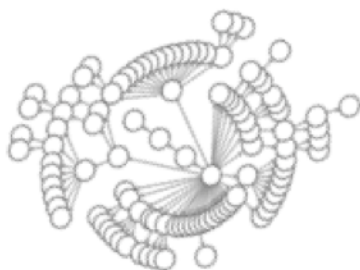


Complicating Factors



Complexity

One component
may rely on 00s
of others



Diversity

40,000 Projects
200MM Classes
400K Components



Volume

Typical Enterprise
Consumes 000s
of Components
Monthly



Change

Typical Component
is Updated 4X per
Year

Component	Version	Release Date
Spring	3.0.0	10 April 2008
Spring	3.1.0	22 March 2009
Spring	3.2.0	22 November 2009
Spring	3.2.1	22 November 2009
Spring	3.2.2	22 November 2009
Spring	3.2.3	22 November 2009
Spring	3.2.4	22 November 2009
Spring	3.2.5	22 November 2009
Spring	3.2.6	22 November 2009
Spring	3.2.7	22 November 2009
Spring	3.2.8	22 November 2009
Spring	3.2.9	22 November 2009
Spring	3.2.10	22 November 2009
Spring	3.2.11	22 November 2009
Spring	3.2.12	22 November 2009
Spring	3.2.13	22 November 2009
Spring	3.2.14	22 November 2009
Spring	3.2.15	22 November 2009
Spring	3.2.16	22 November 2009
Spring	3.2.17	22 November 2009
Spring	3.2.18	22 November 2009
Spring	3.2.19	22 November 2009
Spring	3.2.20	22 November 2009
Spring	3.2.21	22 November 2009
Spring	3.2.22	22 November 2009
Spring	3.2.23	22 November 2009
Spring	3.2.24	22 November 2009
Spring	3.2.25	22 November 2009
Spring	3.2.26	22 November 2009
Spring	3.2.27	22 November 2009
Spring	3.2.28	22 November 2009
Spring	3.2.29	22 November 2009
Spring	3.2.30	22 November 2009



Security - bouncycastle



Sponsored by
DHS National Cyber Security Division/US-CERT

NIST
National Institute of
Standards and Technology

National Vulnerability Database

automating vulnerability management, security measurement, and compliance checking

Vulnerabilities | Checklists | 800-53/800-53A | Product Dictionary | Impact Metrics | Data Feeds | Statistics

Home | SCAP | SCAP Validated Tools | SCAP Events | About | Contact | Vendor Comments

Mission and Overview

NVD is the U.S. government repository of standards based vulnerability management data. This data enables automation of vulnerability management, security measurement, and compliance (e.g. FISMA).

Resource Status

NVD contains:
49412 CVE Vulnerabilities
228 Checklists
221 US-CERT Alerts
2570 US-CERT Vuln Notes
7690 OVAL Queries
38648 CPE Names
Last updated: Fri Feb 03 12:26:42 EST 2012
CVE Publication rate: 11.6

Vulnerability Summary for CVE-2007-6721

Original release date: 03/30/2009
Last revised: 01/20/2011
Source: US-CERT/NIST

Overview

The Legion of the Bouncy Castle Java Cryptography API before release 1.38 (aka 2.5.2), as u related to "a Bleichenbacher vulnerability in simple RSA CMS signatures without signed attrib

Impact

CVSS Severity (version 2.0):
CVSS v2 Base Score: 10.0 (HIGH) (AV:N/AC:L/Au:N/C:C/I:C/A:C) (legend)
Impact Subscore: 10.0
Exploitability Subscore: 10.0
CVSS Version 2 Metrics:
Access Vector: Network exploitable
Access Complexity: Low
**NOTE: Access Complexity scored Low due to Insufficient information
Authentication: Not required to exploit
Impact Type: Allows unauthorized disclosure of information: Allows unauthorized modification

- In the Last Year...
- 6,982 Organizations
- Crypto Library
- Level 10 Flaw
- 3 Years After Fix



Very Real Implications



Security

Cost Per Breach: \$5.5M

Struts Exploit Code

```
http://example.org/struts2app/myaction?foo=%28%23context[%22xwork.MethodAccessor.denyMethodExecution%22]%3D+[...],%20%23_memberAccess[%22allowStaticMethodAccess%22]%3d+[...],%20@java.lang.Runtime@getRuntime%28%29.exec%28%27mkdir%20/tmp/PWND%27%29[...%27meh%27%29]=true
```



Intellectual Property

Litigation, settlements, lost IP

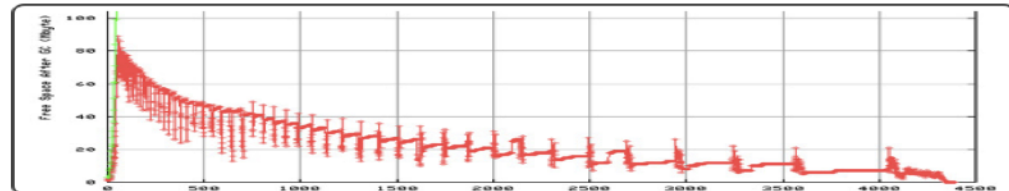


Let's save the GNU!

gpl-violations.org

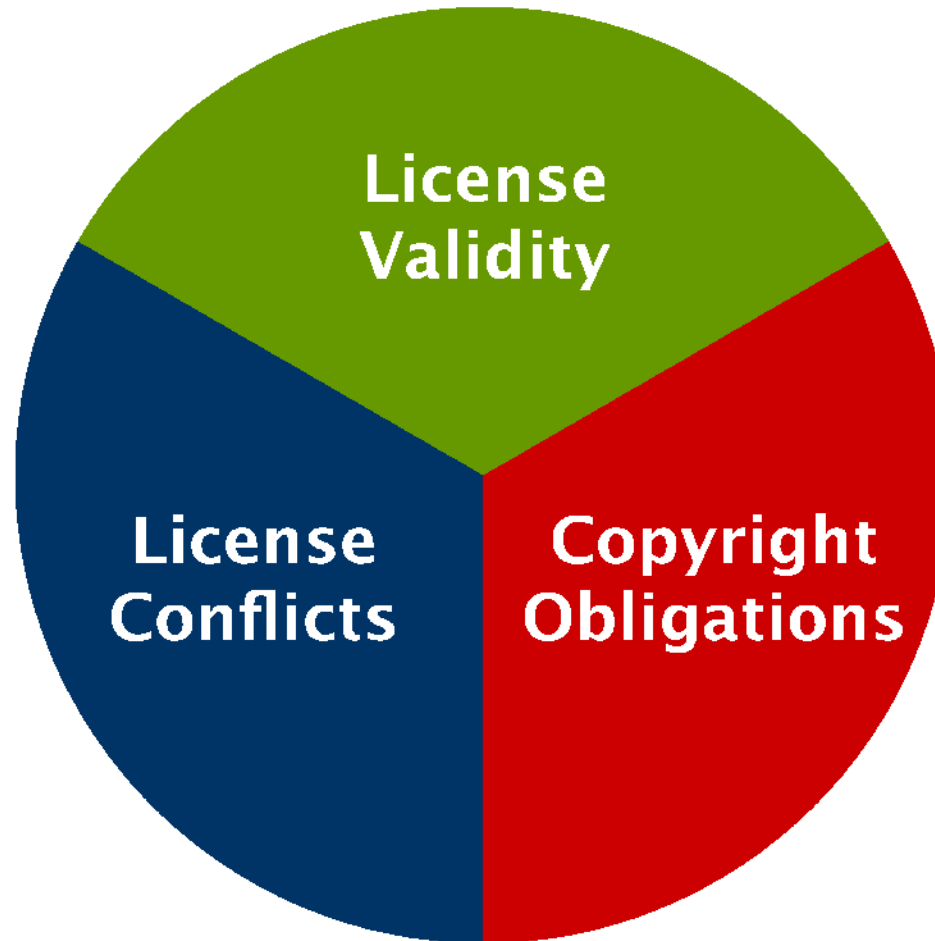
Software Quality

2011 revenue loss: \$26.5 Billion





Intellectual Property - Licensing

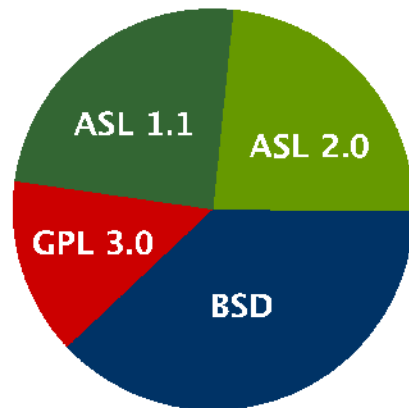




Intellectual Property - Licensing

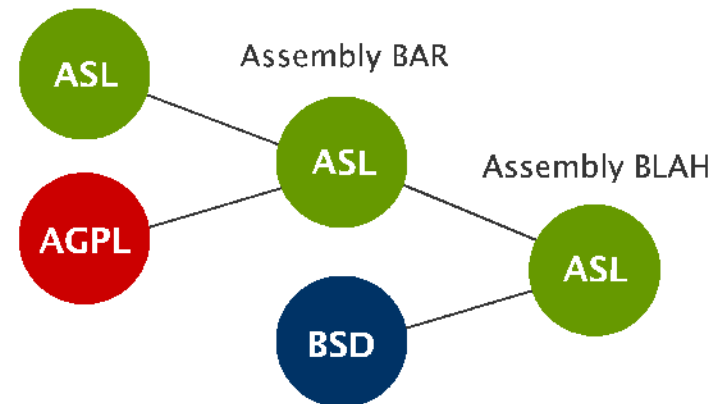


Composite Work FOO



The ASL 1.1 and the GPL are incompatible. FOO cannot be distributed. Period. Under any license.

Assembled Work BLAH



An ASL work cannot subsume an AGPL work (as in BAR).

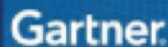


Open Source Benefits

→ Come with Risks



"Above all other considerations, the primary factor in **balancing risk versus reward** from open-source-software (OSS) assets hinges on the **successful execution of an enterprise open-source governance program.**"

Gartner.

A CIO's Perspective on Open Source Software
Mark Driver, Research Vice President

January 2011

→ Introduce Governance



Most Organizations Lack Controls



Control of artifacts in development



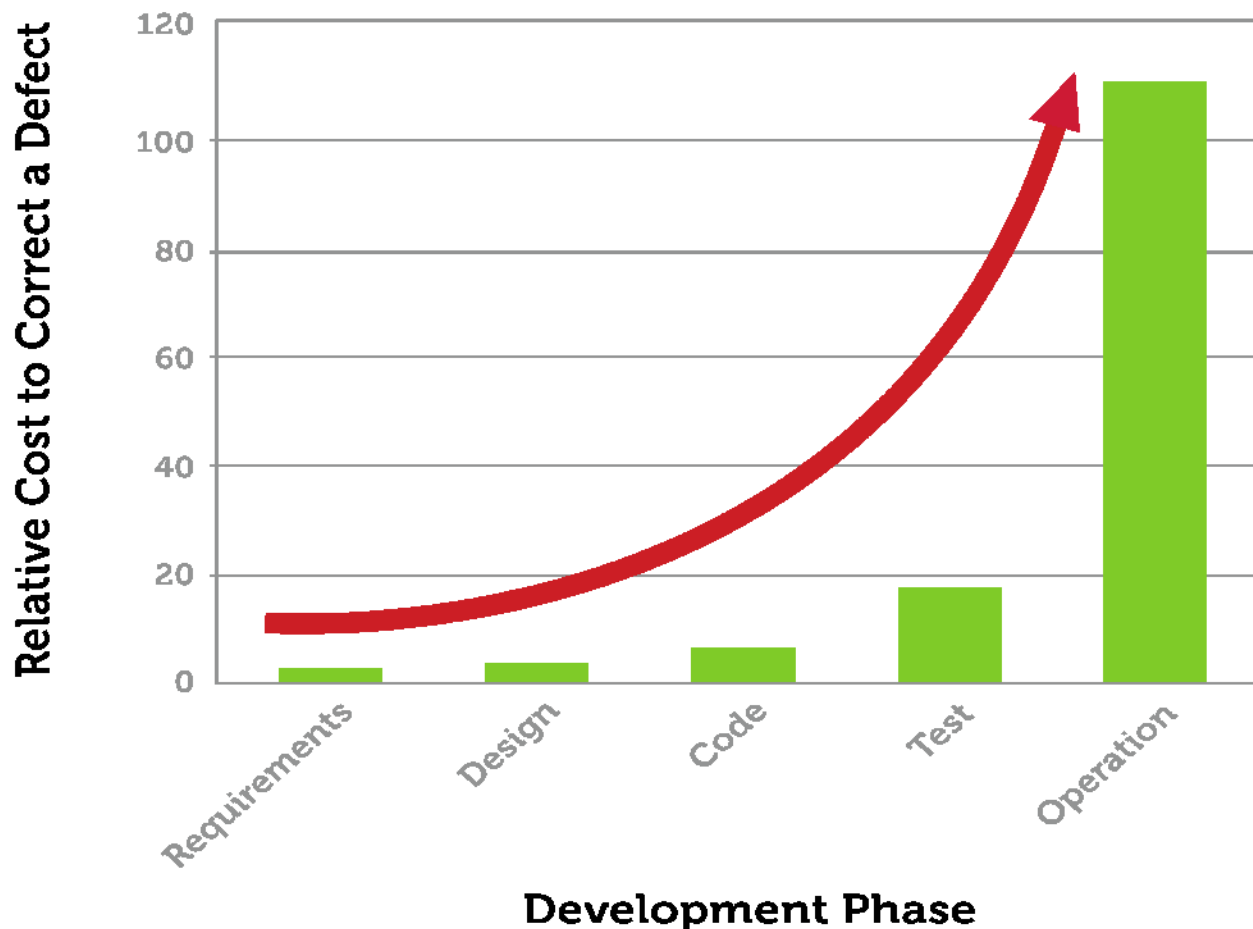
20% We're completely locked down. We can only use approved components.

43% We have some corporate standards, but they aren't enforced.

37% There are no standards. Each developer team choose the components that are best for their project.



Find it Early, Fix it Early



Doh! Obvious right?

But who really does something about it?

How can we deal with this?



- Keep out fingers crossed
- Look the other way
- Hope for the best
- Pretend this flaw does not affect us

But when the \$%!# hits the fan



Your boss will forever be on your back
- or you might have to look for a new job

There is a better way though

- Jenkins already takes care of your build
- Why not let it worry about license and security too?



Insight for CI

→ Let's check it out!



Insight For CI Plugin - Configuration



- Find it in Plugin Manager
- Global Settings
- Insight Build Scan Steps
 - Maven 2/3 or Freestyle jobs
 - Any time – pre, post
 - Configure scan target, packages, failure...



Insight For CI Plugin – What it does



- Creates fingerprint of all artifacts
- Sends to Insight Service
- Service produces matches and provides report
- Plugin downloads, stores and displays report

→ It does not matter how artifacts are created

It is build tool agnostic

Fast – no static code analysis

No privacy/IP problems

Insight For CI Plugin - Results

- Archived per build
 - Summary
 - Components
 - Security Issues
 - License Analysis
-
- Edit license
 - Edit vulnerability
 - Audit log



And once you have results?



Out with the old – in with the new

Security And License Issues



- Use as business reason to
 - upgrade libraries
 - select frameworks and libraries
 - Often hard to justify upgrades for development otherwise
 - Saved time and effort following security announcements and list
 - Reduce effort to sort out license issues
- You get to concentrate on the code and your business value creation



Not running Jenkins/Hudson?

- Then you should install Jenkins and get on with it
- Or wait for Insight for CI for your server
 - Ask us what is coming
- Or try Insight App Health Check



Hang on – what is that?



Insight App Health Check



If you

- Performed the release build already
 - Want to check 3rd party application e.g. app server or war
-
- Download scanner software
 - It creates fingerprints
 - And sends them to the Insight Service
 - Get link to report in an email

<http://www.sonatype.com/Products/Insight-App-Health-Check>



CLM Stages - Inventory



- Track component downloads
- Inventory repositories
- Understand your supply chain



CLM Stages - Analyze



- Key applications
- Internal repositories



CLM Stages - Control



- Policies (security, licensing)
- Establish blacklists
- Implement controls in development



CLM Stages - Monitor



- Maintain an inventory of all component used in production
- Monitor for change and newly discovered vulnerabilities



Requirements for CLM



Precise

Precisely
identify
components,
even when
altered

Actionable

Detailed
security,
quality, and
licensing
information

Complete

Analyzes the
entire
component,
including
dependencies

Update Aware

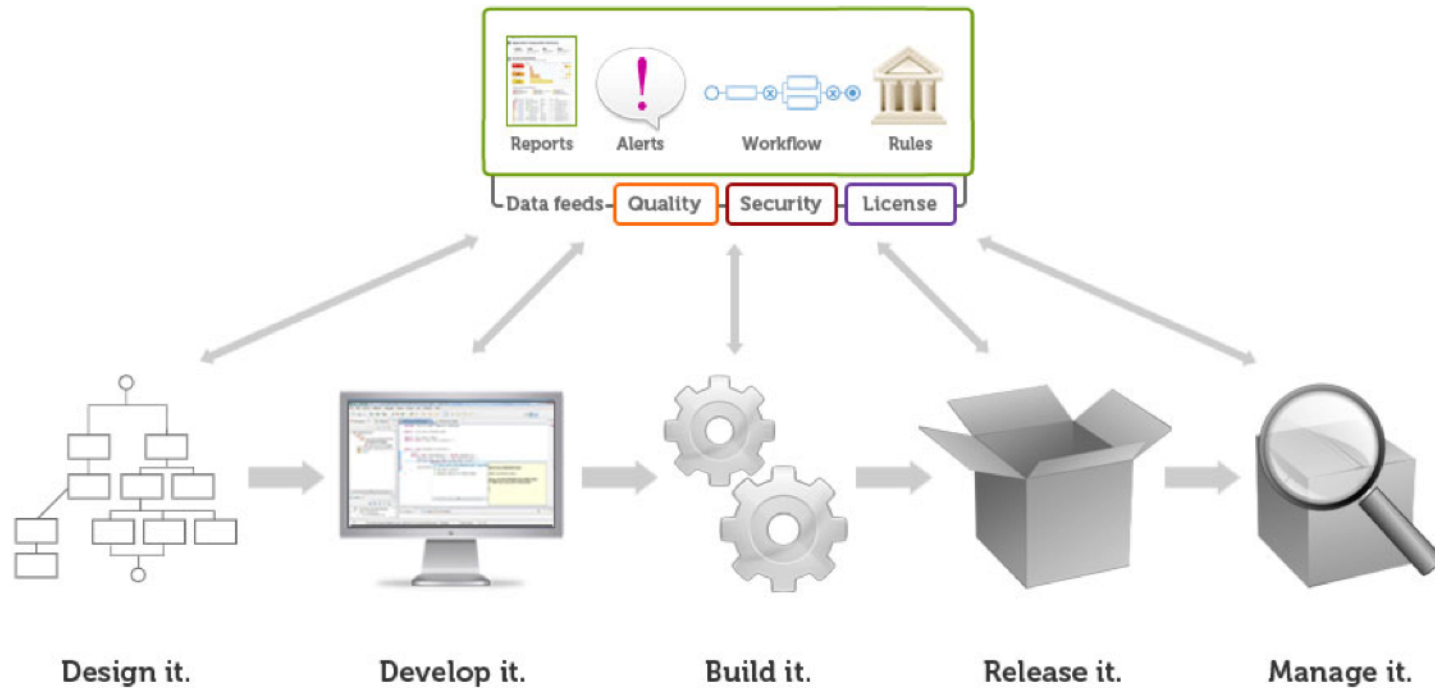
Notifies of new
versions with
detailed update
reason

Integrated

Integrates with
existing software
development
lifecycle tools

Sonatype and CLM

Practical Intelligence Across the Software Lifecycle





What next?

- Just try it...
 - Trivial to install Insight for CI and run it
 - Initial overview reports are free
- Let us know how you like it
- And now Questions, Ideas, Feedback?



Thank You To Our Sponsors

