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

Now

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

Written

Assembled

>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.**

Gartner

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

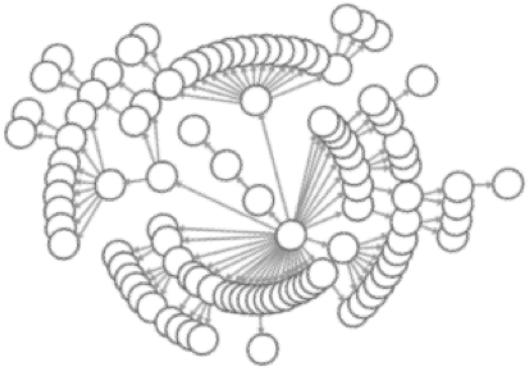
So they don't negatively impact you...

An Immature Ecosystem

No security infrastructure

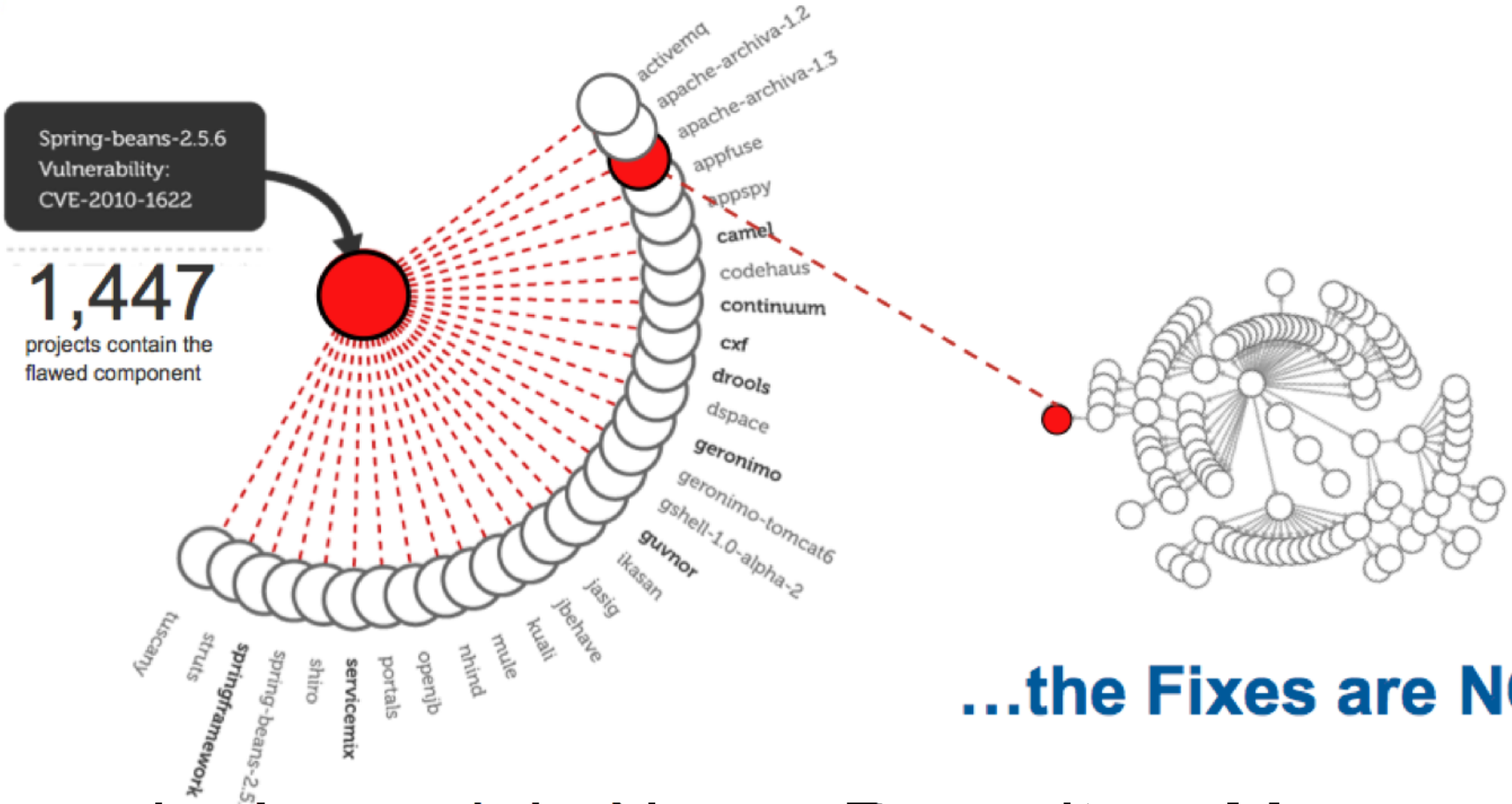
No Update Alerting

No standard licensing mechanisms



…and complex dependencies amplify the problem

Issues Are Viral…



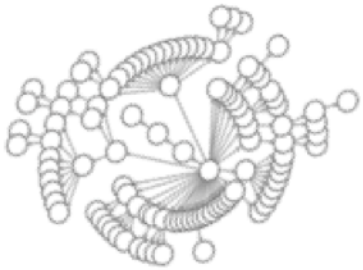
…the Fixes are NOT

Let's see it in Nexus Repository Manager

Complicating Factors

Complexity

One component may rely on 00s of others



Diversity

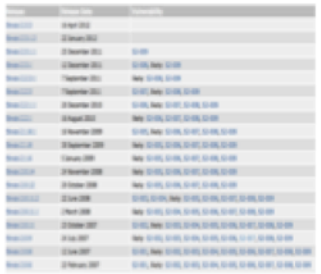
40,000 Projects
200MM Classes
400K Components

Volume

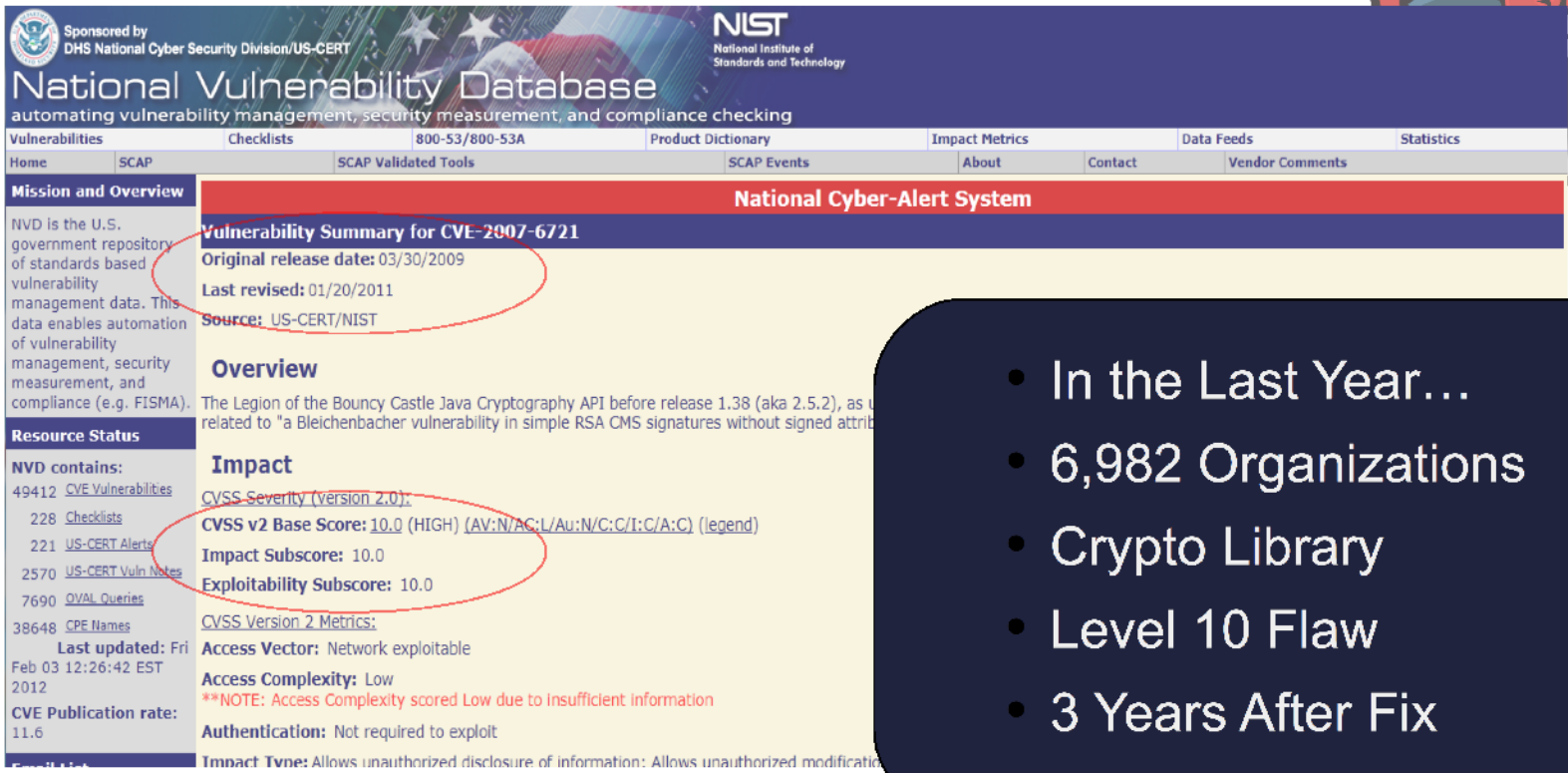
Typical Enterprise Consumes 000s of Components Monthly

Change

Typical Component is Updated 4X per Year

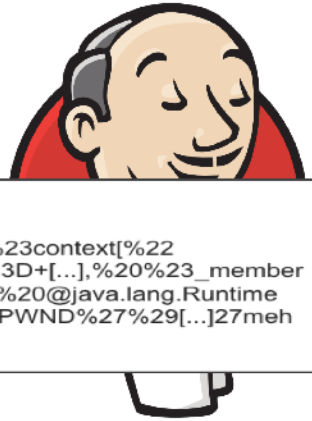


Security - bouncycastle



- 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

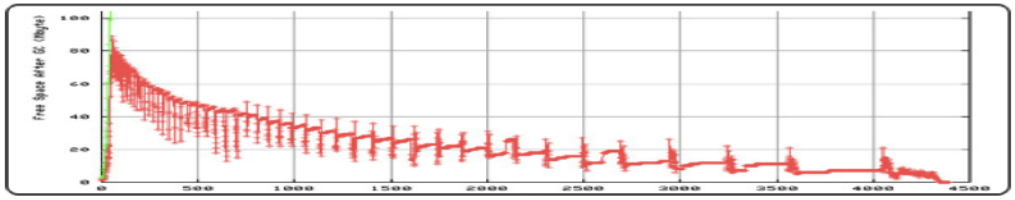
Litigation, settlements, lost IP

Let's save the GNU!

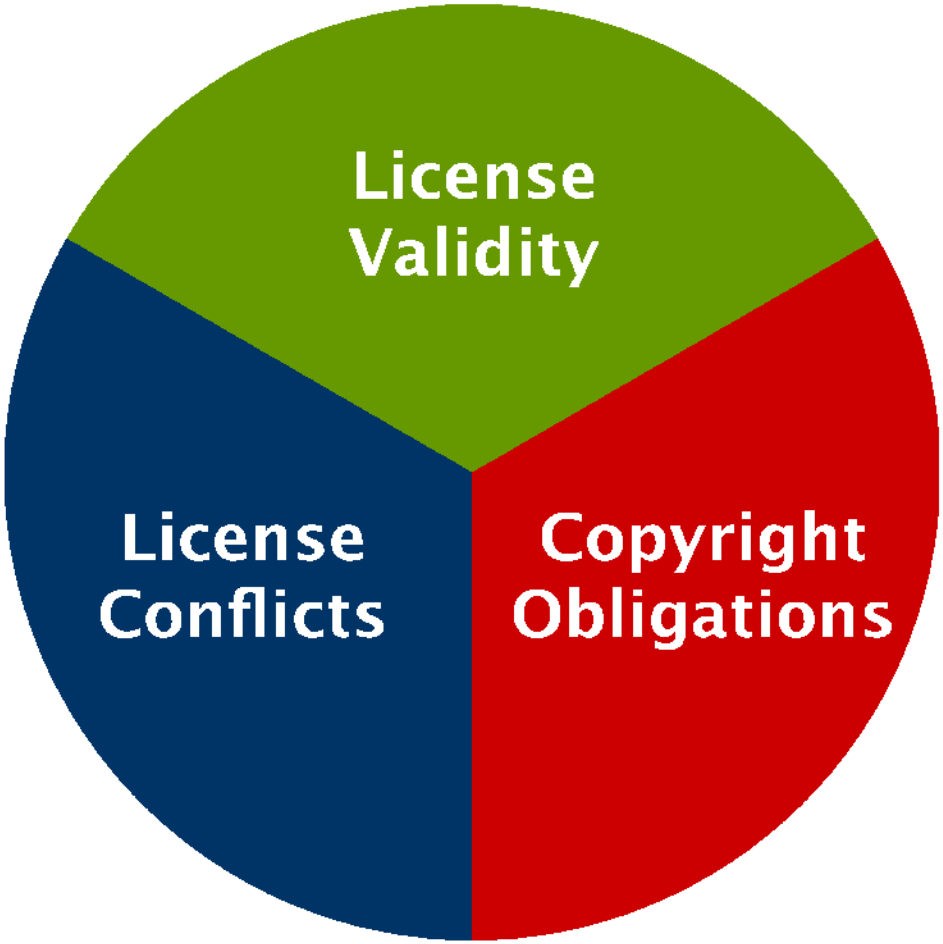
gpl-violations.org

Software Quality

2011 revenue loss: $26.5 Billion

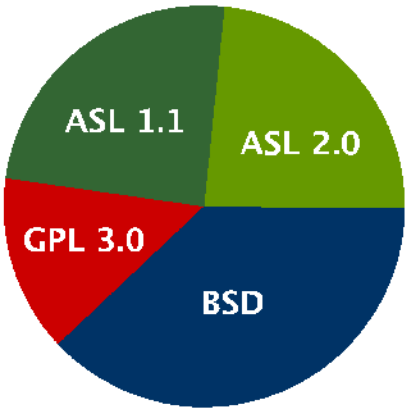


Intellectual Property - Licensing



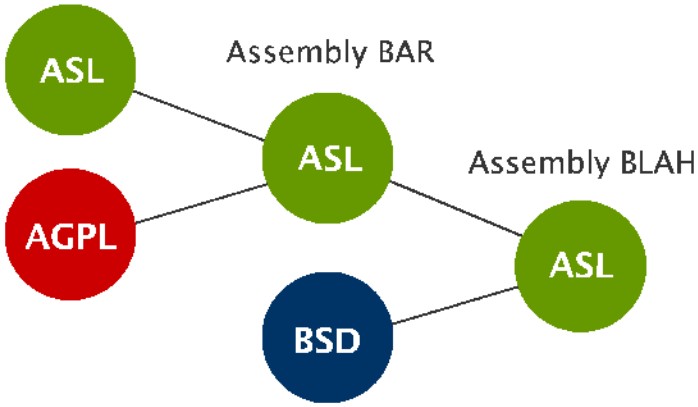
Intellectual Property - Licensing

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

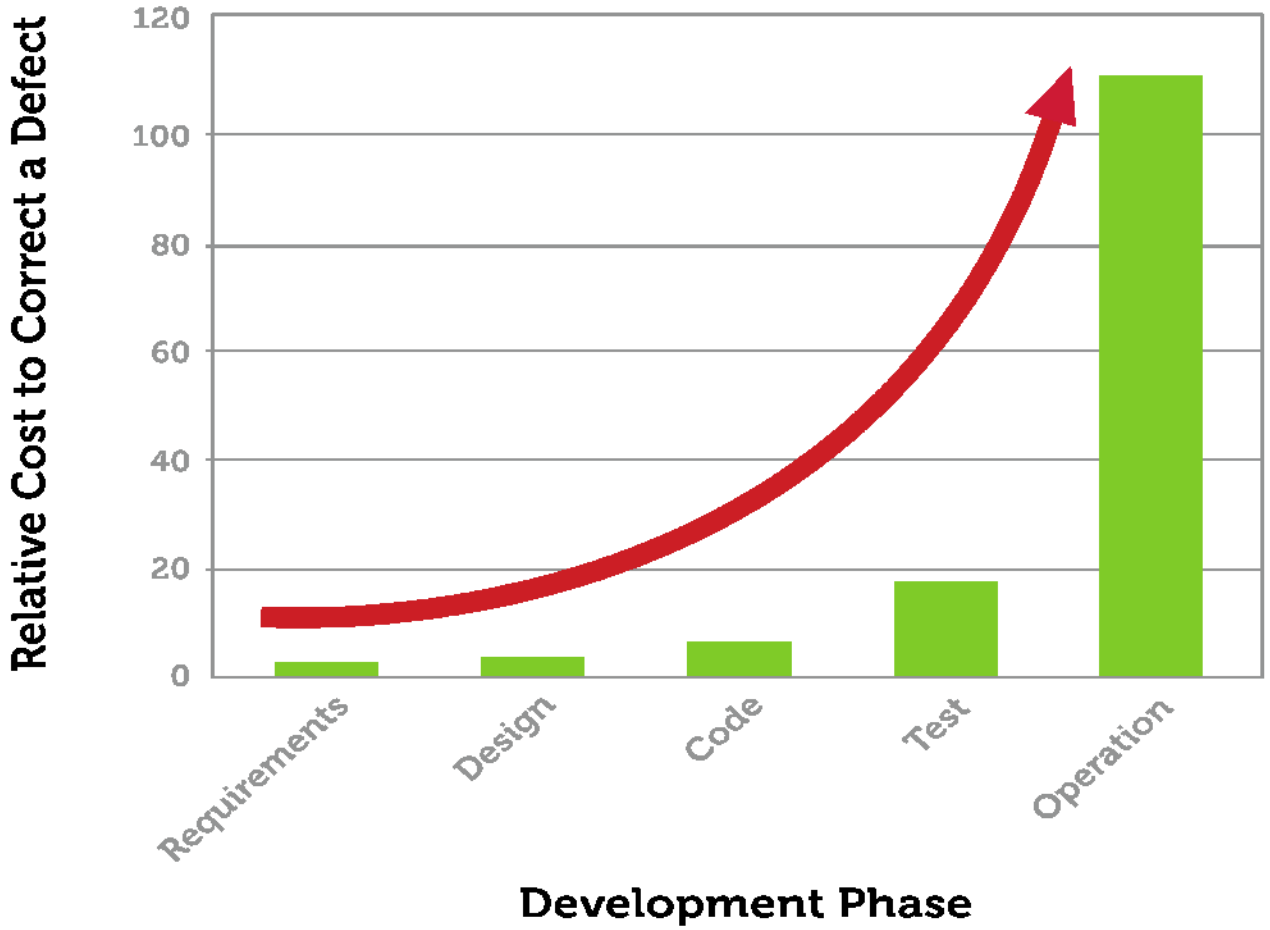
20% We're completely locked down. We can only use approved components.

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.

2012 Sonatype survey of 2,550 developers, architects, and managers

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

Inventory → Analyze → Control → Monitor

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

CLM Stages - Analyze

Inventory → **Analyze** → Control → Monitor

- Key applications
- Internal repositories

CLM Stages - Control

Inventory → Analyze → **Control** → Monitor

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

CLM Stages - Monitor

Inventory → Analyze → Control → **Monitor**

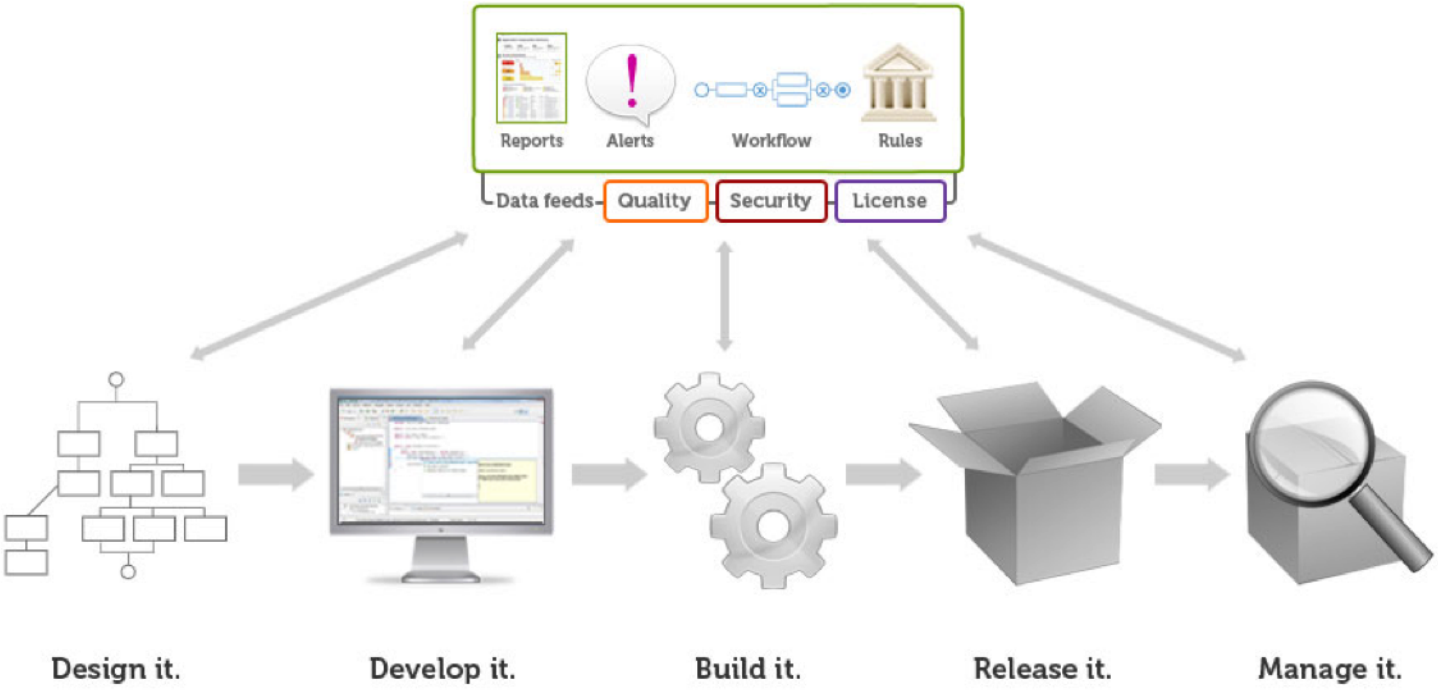
- Maintain an inventory of all component used in production
- Monitor for change and newly discovered vulnerabilities

Requirements for CLM

Precise	Actionable	Complete	Update Aware	Integrated
Precisely identify components, even when altered	Detailed security, quality, and licensing information	Analyzes the entire component, including dependencies	Notifies of new versions with detailed update reason	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

CLOUDANT

Sonatype

SendGrid
Email Delivery. Simplified.

YAHOO!