

# **Stop the Software Architecture Erosion**



**Bernhard Merkle**

**Central Research & Development**

**Software-Engineering**

**SICK-AG Waldkirch**

**[mailto: Bernhard.Merkle@gmail.com](mailto:Bernhard.Merkle@gmail.com)**

**Contact on [linkedin.com](#) or [xing.com](#)**

# Disclaimer

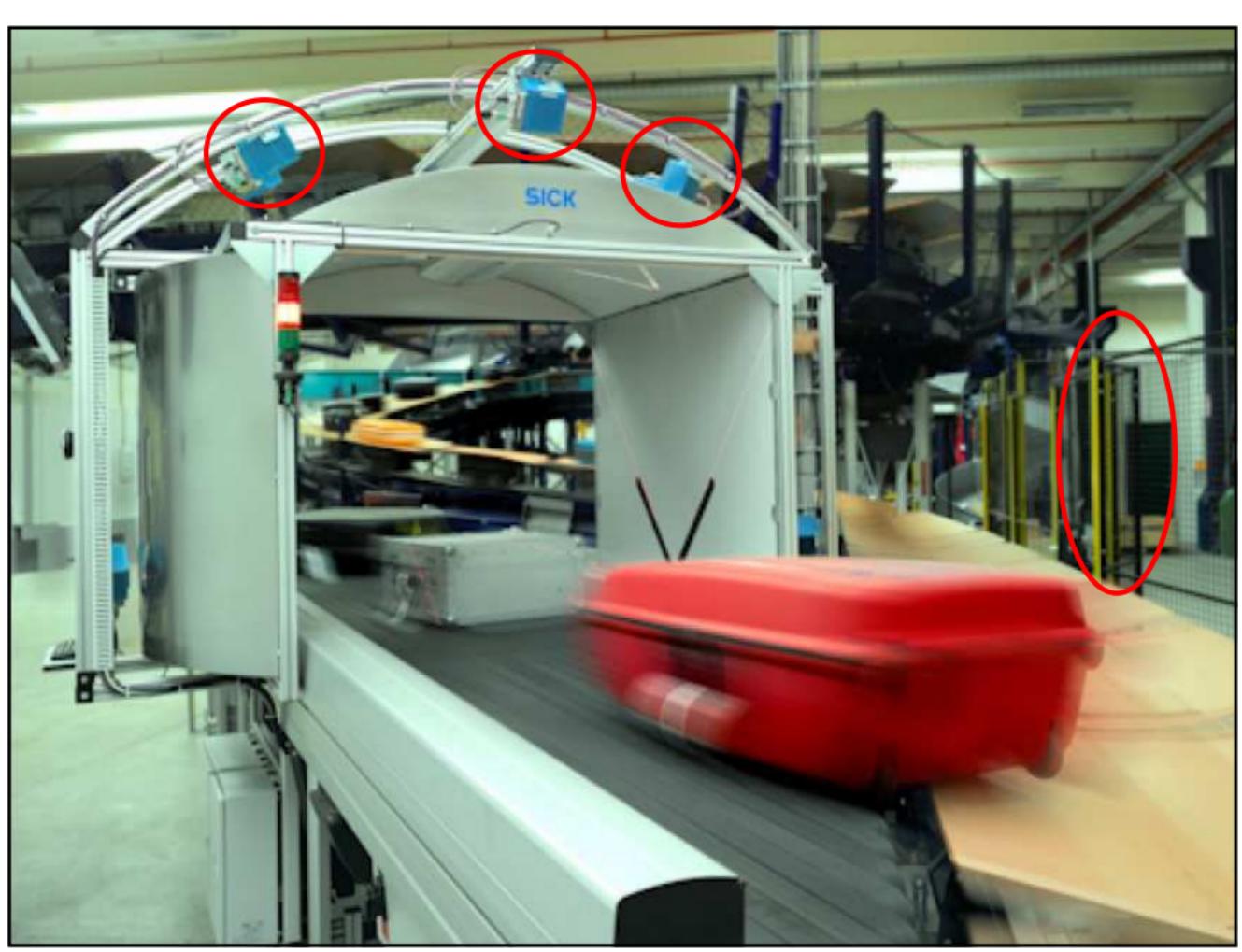
SICK

I am a **SICK** Guy

We will be X-raying Software

Do not shoot me,  
this is a contribution,not criticism !





SICK

# Google: Street View

SICK



# Introduction / Overview

## Levels of Static Analysis

- Code, Design, Architectural
- Examples

## Architectural Analysis

- Use Cases
- Tool Support
- Examples
- Pros/Cons

# Levels of Static Analysis:

---

## Micro-Level

- Code
- =, ==, {}

## Macro-Level

- Class-Design, API-Design
- by reference, Exception-Handling, equals/hashCode/op=

## Architecture-Level:

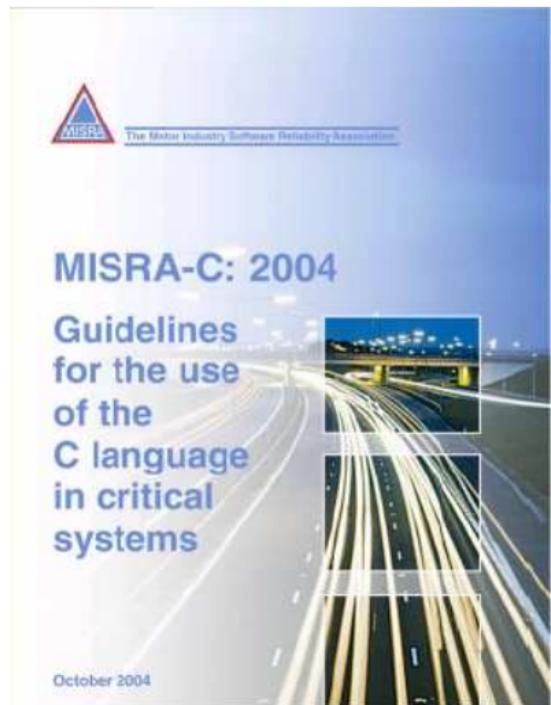
- Layers, Subsystems, Components, Interfaces
- Coupling, Dependency, etc...

# Micro-Level: MISRA-C 2004

```
#define MAX(a, b) ((a) > (b)) ? (a) : (b)
/* ... */
z = MAX( i++, j );

x = b[i] + i++;

x = func( i++, i );
```



# Macro-Level: Design+Coding Guidelines

## Effective C++ Second Edition

50 Specific Ways to Improve Your Programs and Designs

Scott Meyers



Conform  
new ISO  
C++ stan

ADDISON-WESLEY PROFESSIONAL

Effective SOFTWARE DEVELOPMENT SERIES  
Scott Meyers, Consulting Editor

## Effective C#

50 Specific Ways to Improve Your C#

Joshua Bloch

## Effective Java® Second Edition



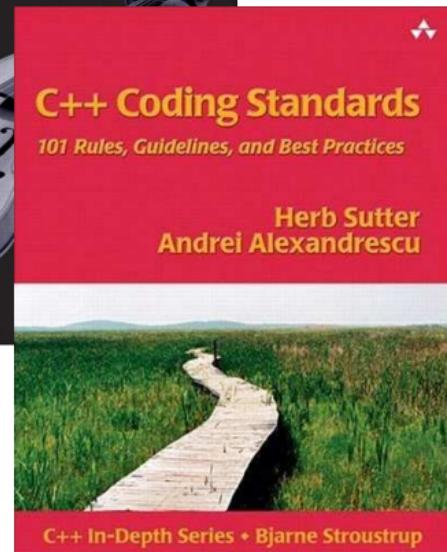
from the Source



## C++ Coding Standards

101 Rules, Guidelines, and Best Practices

Herb Sutter  
Andrei Alexandrescu



# Samples for Effective Rules

---

## C++

- Prefer `const` and `inline` to `#define`
- Use same form in corresponding uses of `new` and `delete`
- Declare ctor and assignment op for classes with dyn memory
- Strive for class interfaces that a minimal and complete
- Choose carefully betw. fuction overloading / parameter default

## Java

- Eliminate obsolete object references
- Always override `hashCode` when you override `equals`
- Minimize mutability
- Favor composition over inheritance
- Do not use raw types in new code

# Tool Example Java: findbugs

The screenshot shows the FindBugs application window. At the top, there's a menu bar with 'Datei', 'Ansicht', 'Einstellungen', and 'Hilfe'. Below the menu is a toolbar with four buttons: 'Nach Klasse', 'Nach Paket', 'Nach Fehlertyp', and 'Übersicht' (Overview). The 'Übersicht' button is highlighted.

The main pane displays a list of code smells, each with a red icon and a brief description:

- Eq: Covariant equals() (1)
- HE: Equal objects must have equal hashcodes (4)
- IS2: Inconsistent synchronization (15)
- LI: Unsynchronized Lazy Initialization (1)
- MF: Masked Field (7)
- MS: Mutable static field (70)
- NN: Naked notify in method (8)
- NP: Null pointer dereference (1)
- RC: Suspicious reference comparison (1)
- SBSC: String concatenation in loop using + operator (10)
- SIC: Inner class could be made static (23)
- Se: Incorrect definition of Serializable class (12)
- SnVI: Serializable class with no Version ID (35)
- UICE: Unsigned control flow (1)

Below this list, there's a navigation bar with three tabs: 'Einzelheiten' (selected), 'Quelltext', and 'Anmerkungen'. The 'Einzelheiten' tab contains the following content:

## Field should be package protected

A mutable static field could be changed by malicious code or by accident. The field could be made package protected to avoid this vulnerability.

# Software, Architecture, Erosion

---



# Software-Architecture: Definitions

IEEE 1471-2000:

The fundamental **organization** of a system,  
embodied in its **components**,  
their **relationship** to each other and the environment,  
and the **principles** governing its **design** and **evolution**.

Kruchten: captured in

- *Software Architecture Document*
- *Software Design Guidelines*

# Erosion always happens

- Prototypes become products
- Hacks/Workarounds
- Lack of understanding “should architecture”
- Time pressure

# Architectural Erosion

*“Sometimes the developers manage to maintain this purity of design through the initial development and into the first release.*

*More often something goes wrong.  
The software starts to rot like a piece of bad meat”.*

Uncle Bob: “Agile Software Development”

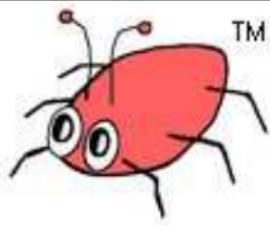


There is/was a plan

# X-raying Software...

Is there a architecture ?

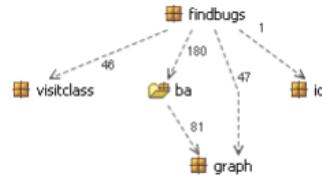
Is there a erosion ?



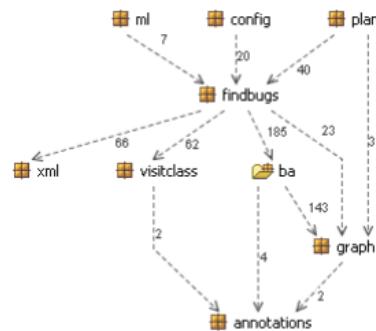
Do AA-Tool work well ?

Do codelevel lints care about architecture ?

# Findbugs: the first years

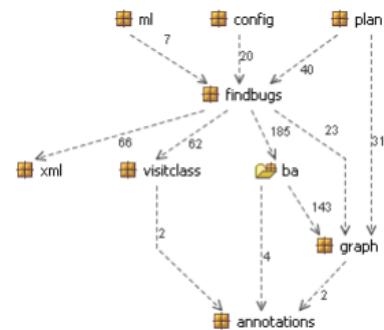


0.7.2  
(03/2004)

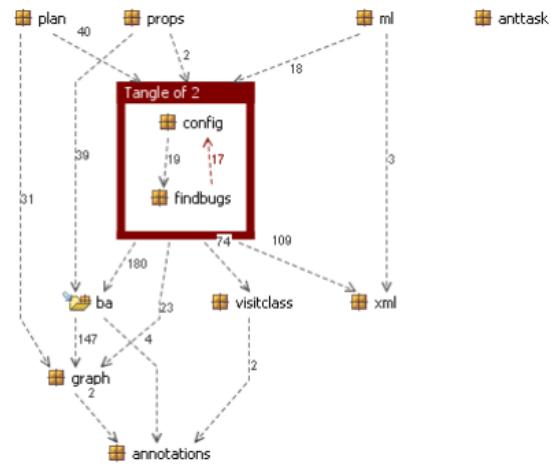


0.8.6  
(10/2004)

# Findbugs: the first years

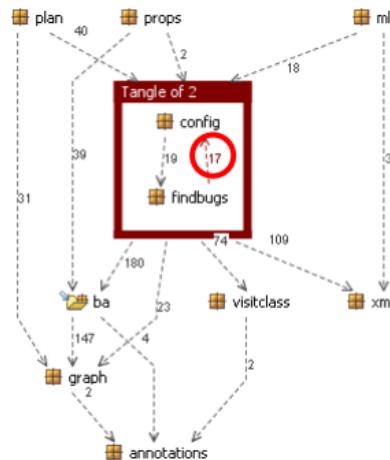


0.8.6  
(10/2004)



0.8.7  
(05/2005)

# Findbugs 0.8.7: Architectural Analysis



**Log Messages - http://findbugs.googlecode.com/svn**

From: 01.03.2005 To: 01.05.2005

Messages, authors and paths

Revision	Actions	Author	Date	Message
3716	modified	daveh0	22:20:33, Sonntag, 6. März 2005	Eliminated some inadvertent autoboxing.
3715	modified	daveh0	22:11:19, Sonntag, 6. März 2005	Added getUserPreferencesFile(). Modified to use IO.writeF
3714	modified	daveh0	22:10:25, Sonntag, 6. März 2005	Initial checkin
3713	modified	daveh0	21:42:31, Sonntag, 6. März 2005	Fixed copying of findbugs.jar and bcel.jar.
3712	modified	daveh0	21:37:12, Sonntag, 6. März 2005	Fix NPE on compilation now that DetectorFactory no longer
3711	modified	daveh0	21:33:02, Sonntag, 6. März 2005	Temporary hack
3710	modified	daveh0	21:31:16, Sonntag, 6. März 2005	Detectors are no longer enabled/disabled by setting an ena
3709	modified	daveh0	21:14:05, Sonntag, 6. März 2005	cleared up imports, added serialVersionUID
3708	modified	daveh0	21:13:07, Sonntag, 6. März 2005	Added serialVersionUID
3707	modified	daveh0	21:10:09, Sonntag, 6. März 2005	Add serialVersionUID
3706	modified	daveh0	19:45:55, Sonntag, 6. März 2005	Modified clone() to call super.clone(), and to use generic ty

Detectors are no longer enabled/disabled by setting an enabled field in the DetectorFactory. Instead, an instance of UserPreferences records which detectors are enabled/disabled.

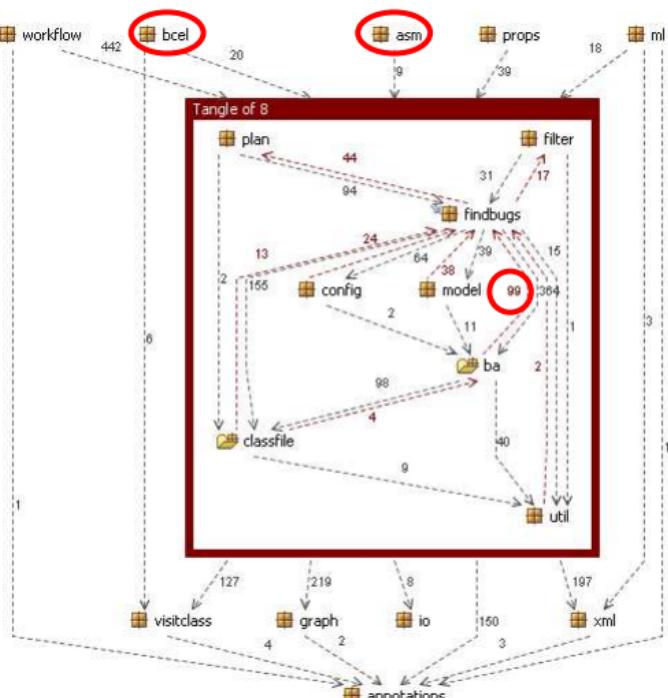
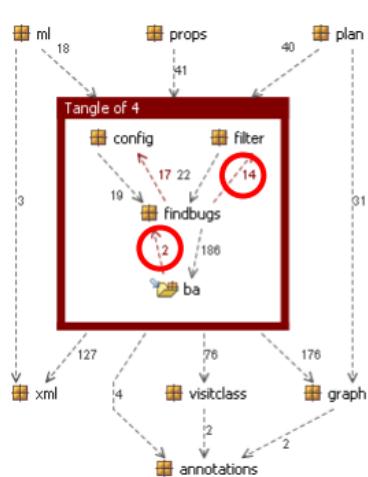
Action	Path	Copy from path	Revision
Modified	/trunk/findbugs/src/java/edu/umd/cs/findbugs/DetectorFactory.java		
Modified	/trunk/findbugs/src/java/edu/umd/cs/findbugs/DetectorFactoryCollection.java		
Modified	/trunk/findbugs/src/java/edu/umd/cs/findbugs/FindBugs.java		
Modified	/trunk/findbugs/src/java/edu/umd/cs/findbugs/config/ProjectFilterSettings.java		
Modified	/trunk/findbugs/src/java/edu/umd/cs/findbugs/config/UserPreferences.java		
Modified	/trunk/findbugs/src/java/edu/umd/cs/findbugs/gui/AnalysisRun.java		
Modified	/trunk/findbugs/src/java/edu/umd/cs/findbugs/gui/ConfigureDetectorsDialog.java		
Modified	/trunk/findbugs/src/java/edu/umd/cs/findbugs/gui/FindBugFrame.java		

Showing 438 revision(s), from revision 3647 to revision 4084 - 1 revision(s) selected.

Hide unrelated changed paths    Stop on copy/ rename    Include merged revisions

Show All   Next 100   Refresh   Statistics   Help   OK

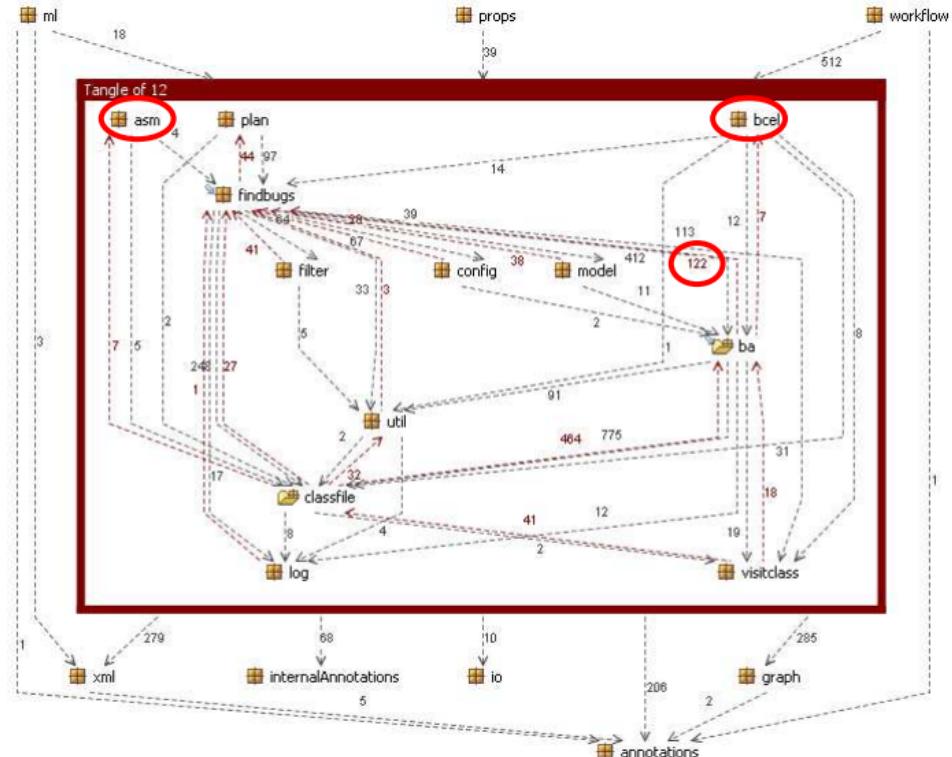
# Tangle increase...



0.8.8  
(05/2005)

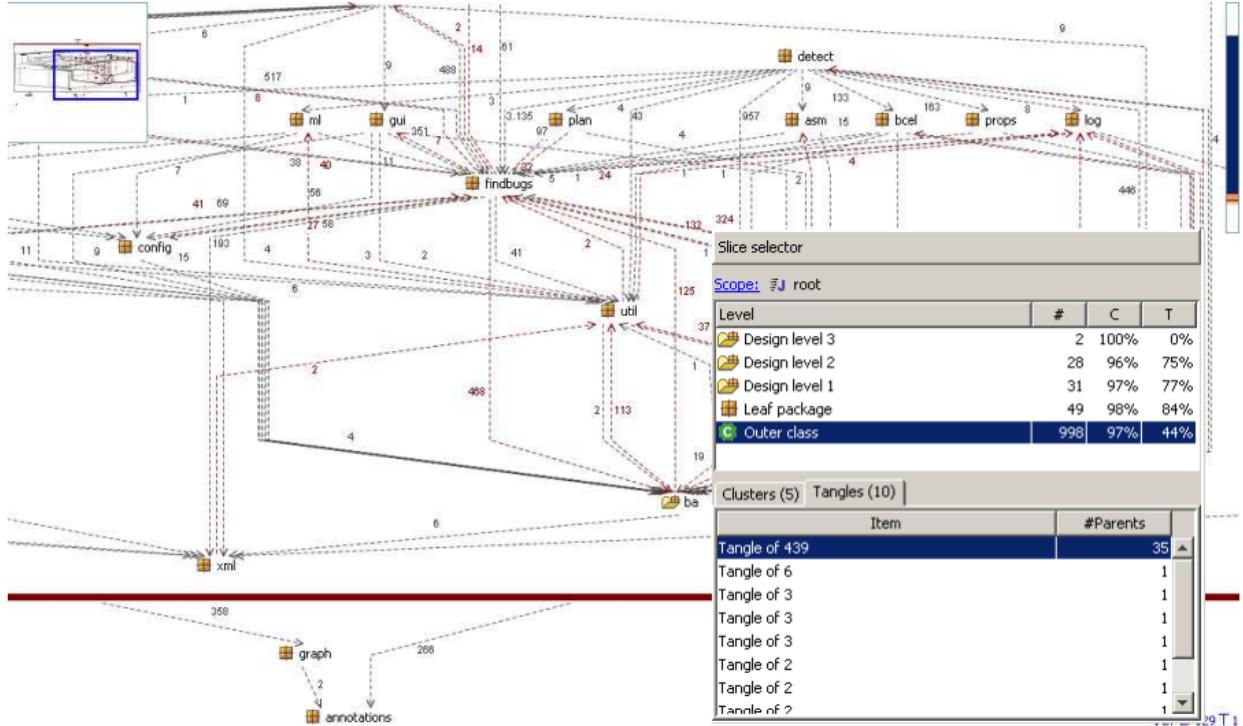
1.0.0  
(06/2006)  
Bernhard Merkle „Stop the Software Architecture Erosion“  
Page: 22

# Tangle increase...



1.3.0

# ONE BIG Tangle...

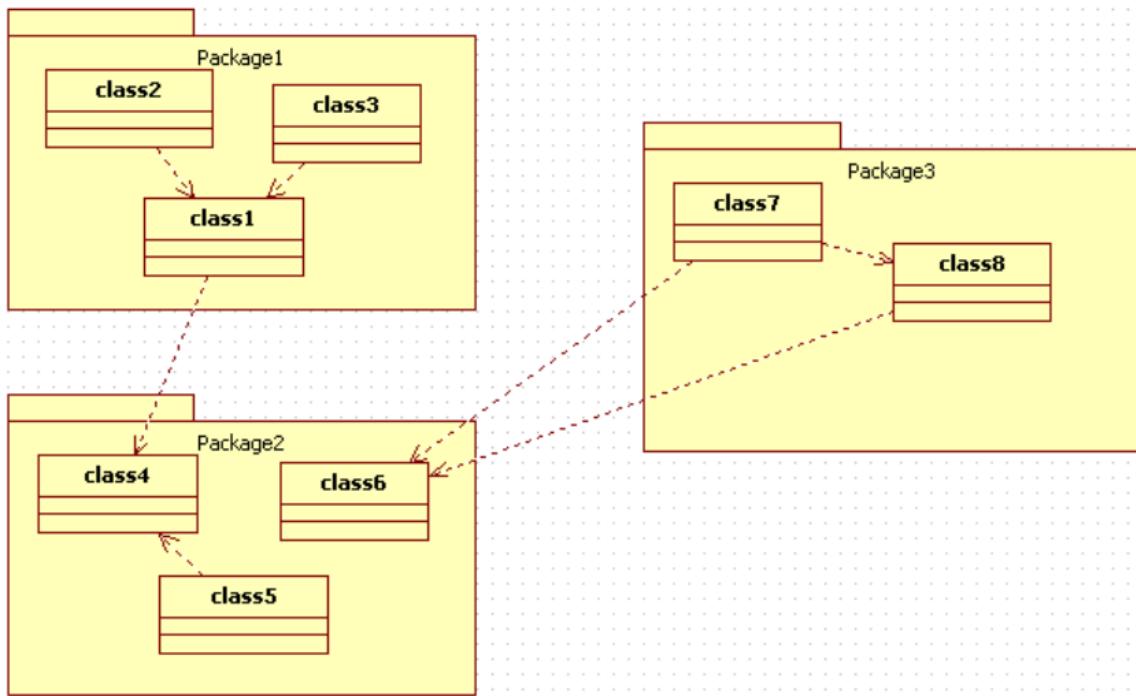


1.3.8 (03/2009)

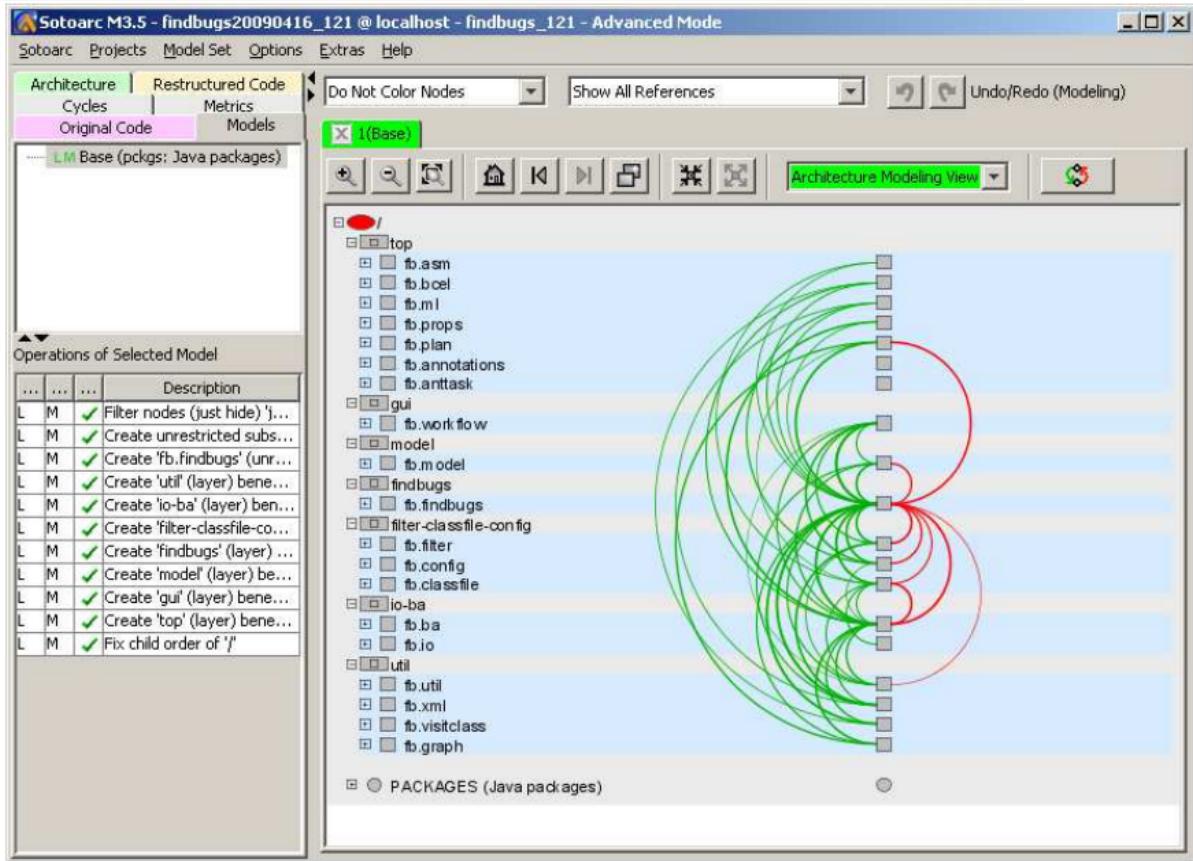
# Still manageable ?



# Anti-Pattern: Findelkind (foundling)



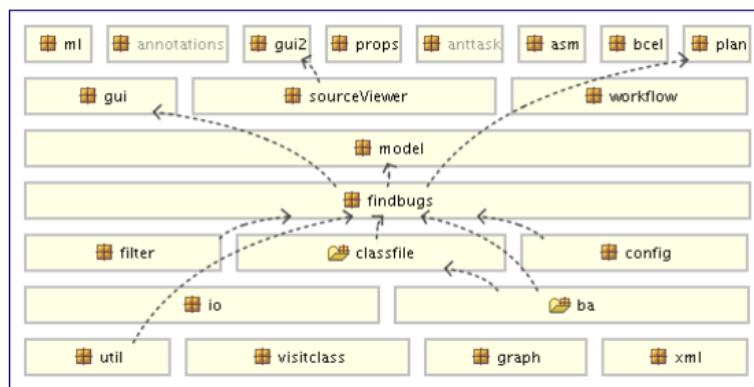
# Modeling Subsystems:



# Fixing Architectural Violations

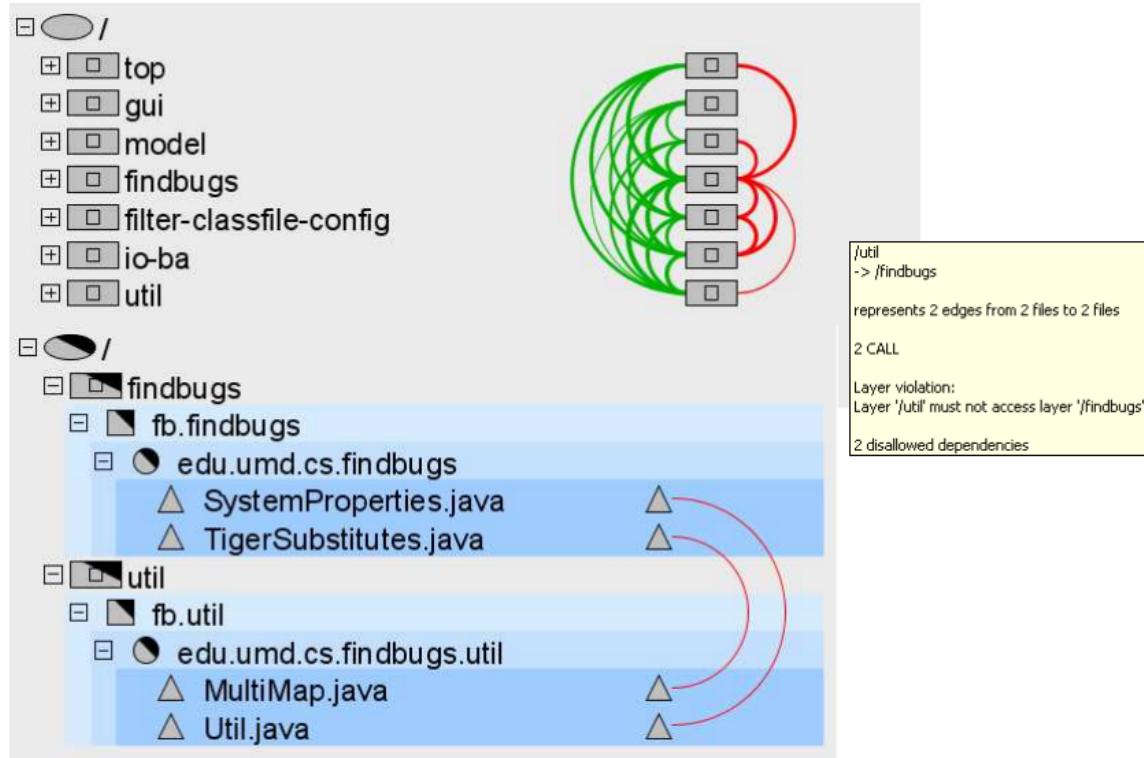


Diagram 1



Description: Subsystem breakout for 'edu.umd.cs.findbugs'

# Fixing Architectural Violations



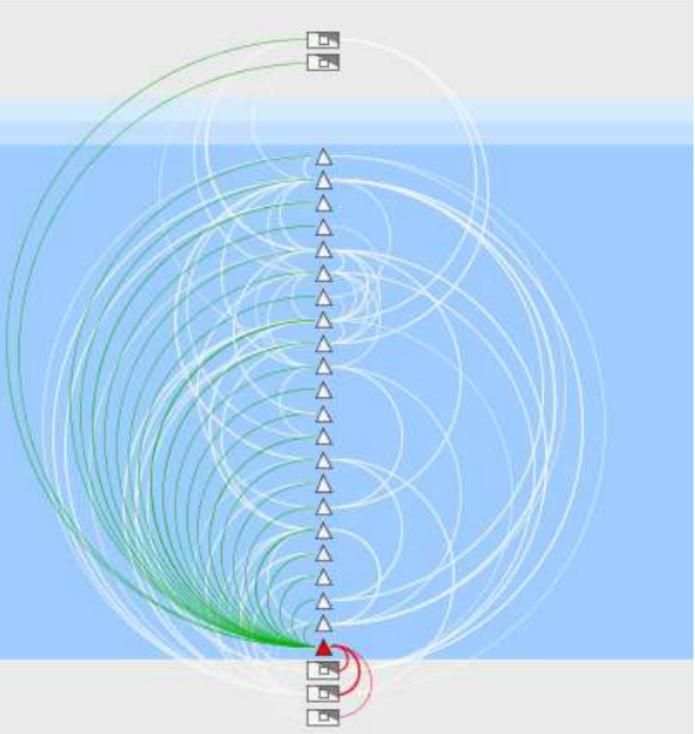
# Fixing Architectural Violations

B:/findbugs/src/findbugs-1.2.1/src/java/edu/umd/cs/findbugs/util/Util.java

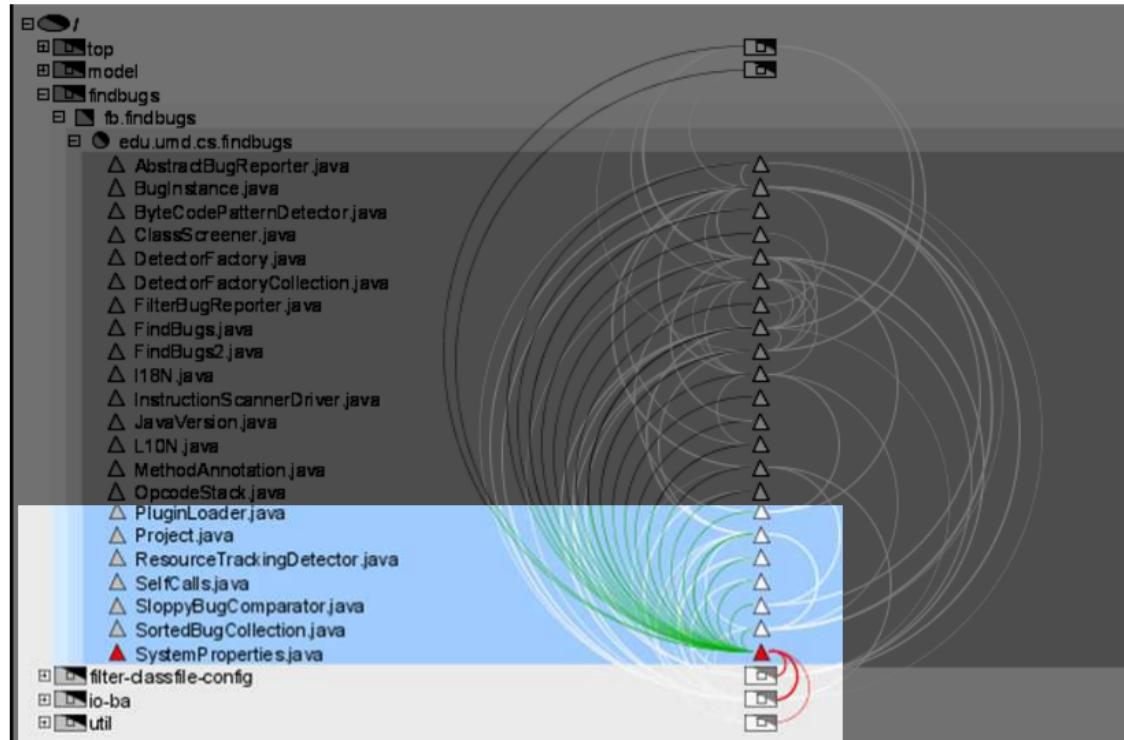
ID	ID	fromName	fromKind	ID	fromFile	ID	toName	toKind	ID	toFile	referenceType
PO	SY	{init_Util_attributes}	METHOD	SF	Util.java	SY	getBoolean	METHOD	SF	SystemProperties.java	CALL

```
39 import edu.umd.cs.findbugs.annotations.CheckForNull;
40
41 /**
42 * @author pugh
43 */
44 public class Util {
45     public static final boolean LOGGING = SystemProperties.getBoolean("findbugs.shutdownLogging");
46
47     public static void runLogAtShutdown(Runnable r) {
48         if (LOGGING) Runtime.getRuntime().addShutdownHook(new Thread(r));
49     }
50 }
51
52     public static int nullSafeHashCode(@CheckForNull Object o) {
53         if (o == null) return 0;
54         return o.hashCode();
55     }
56     public static <T> boolean nullSafeEquals(@CheckForNull T o1, @CheckForNull T o2) {
57         if (o1 == o2) return true;
```

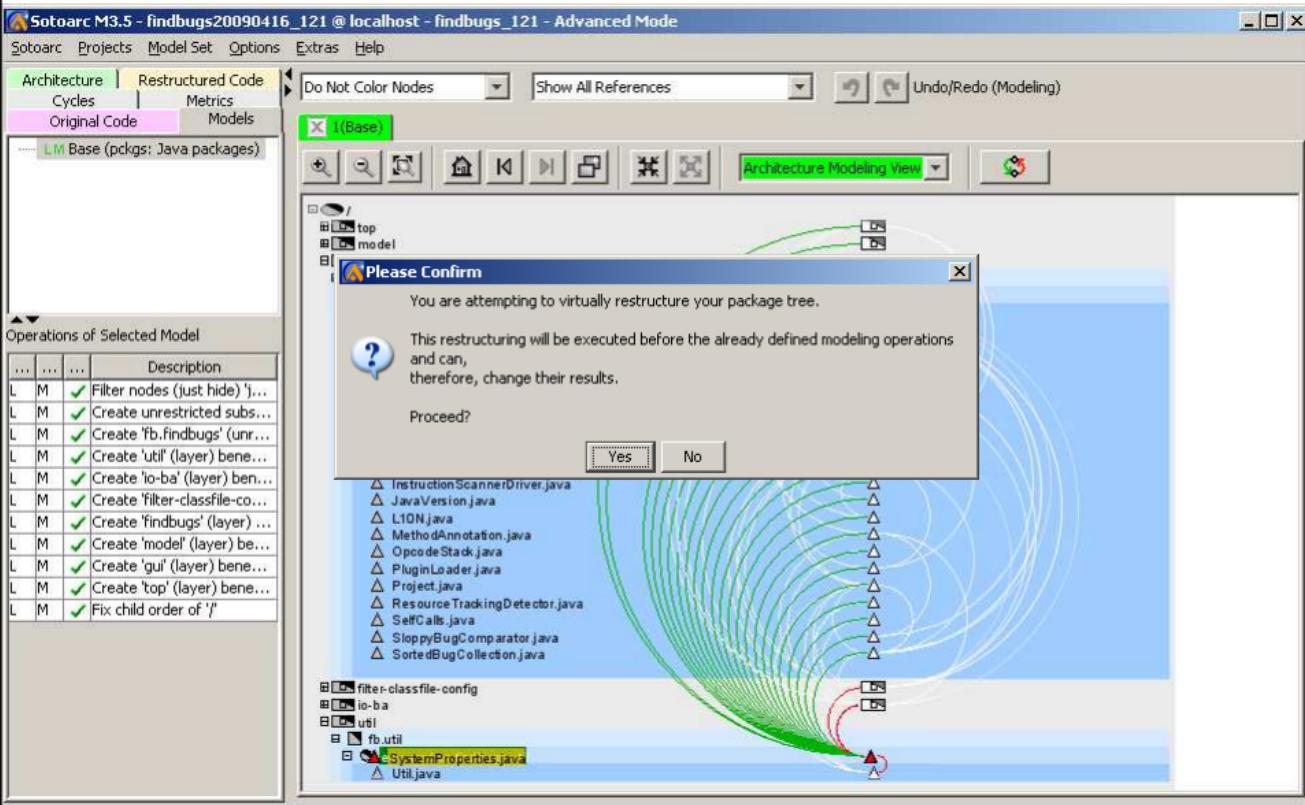
# Fixing Architectural Violations



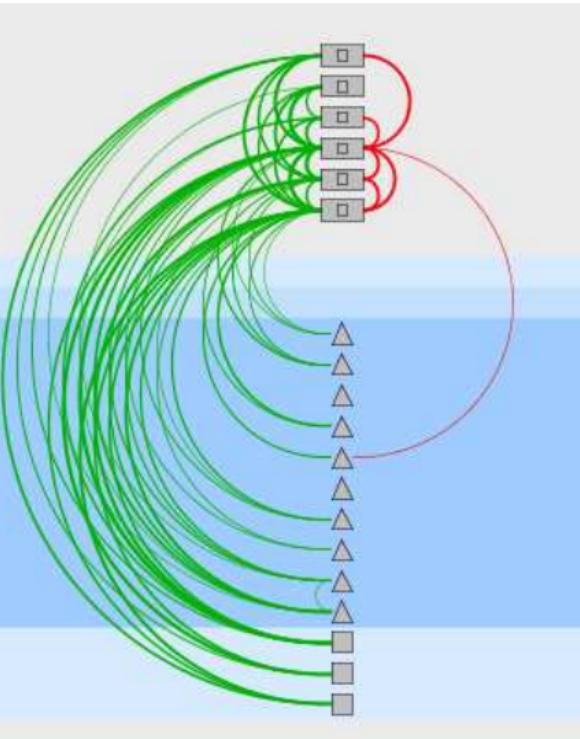
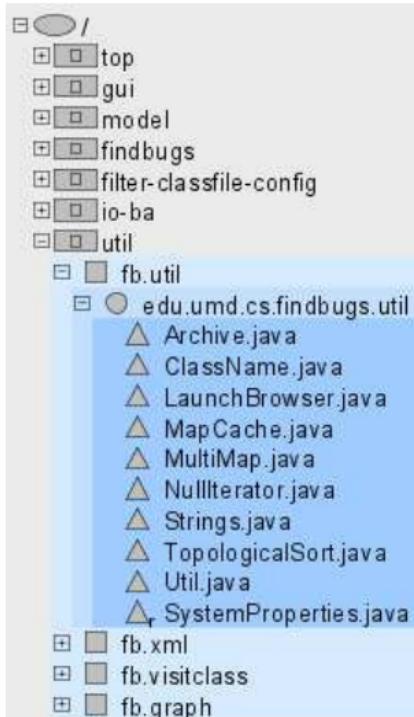
# Fixing Architectural Violations



# Fixing Architectural Violations



# Fixing Architectural Violations



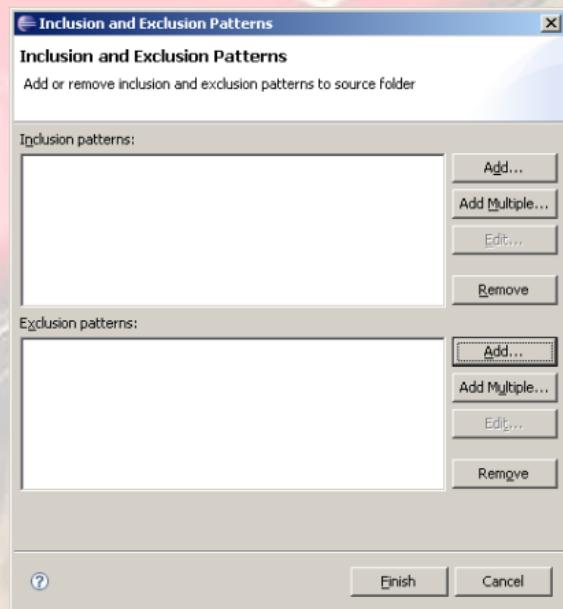
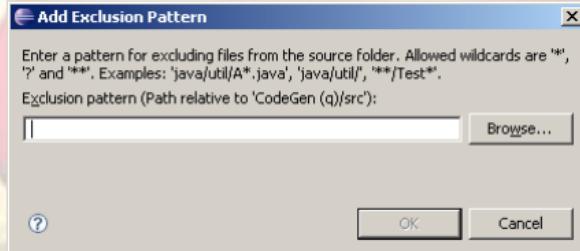
# Tools for Architecture-Analysis

- Sotograph
- Bauhaus
- Structure101
- SonarJ
- Lattix
- Klocwork, Coverity
  
- <http://code.google.com/p/architecturerules/>
- ODASA, CodeCrawler, Codecity, SeeSoft, XRadars, XDepend, SonarSource, Kalistick, Sqoring, ...

**[http://se-radio.net/  
episode-115-architecture-analysis](http://se-radio.net/episode-115-architecture-analysis)**

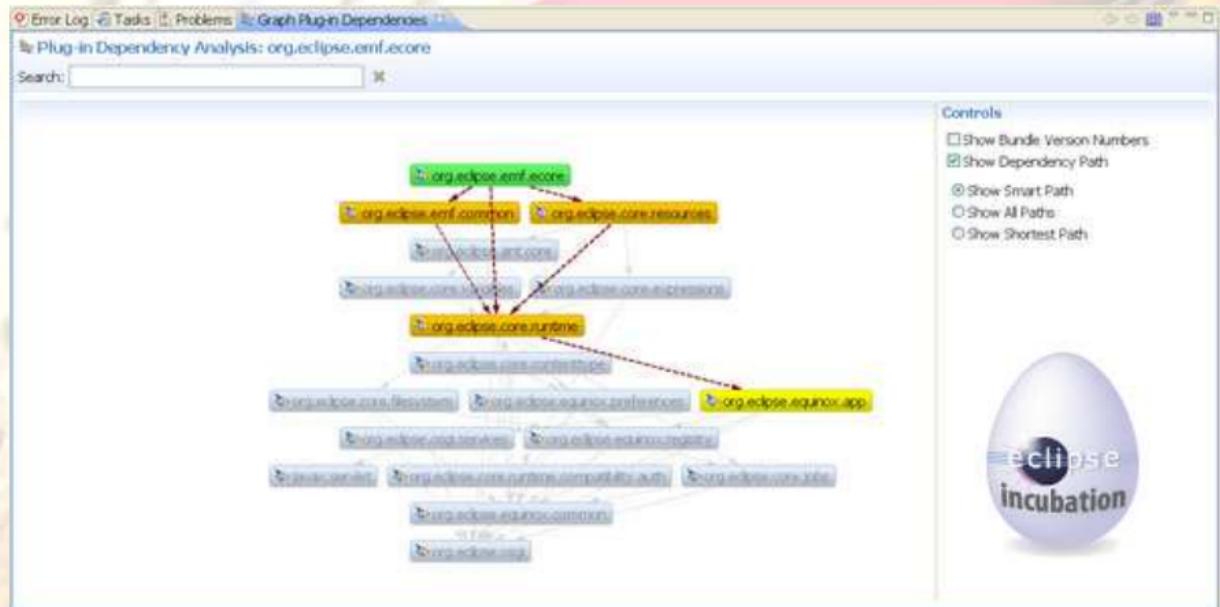
# Basic Approaches

- Your makesystem...
- makedepend, jdepend
- Eclipse (Java Build Path)
- OSGI (Dependencies)



# Basic Approaches

## PDE Dependency Visualization



# Missing in basic approaches:

Architecture Analysis (Deviation)

Drill-Down + Aggregation

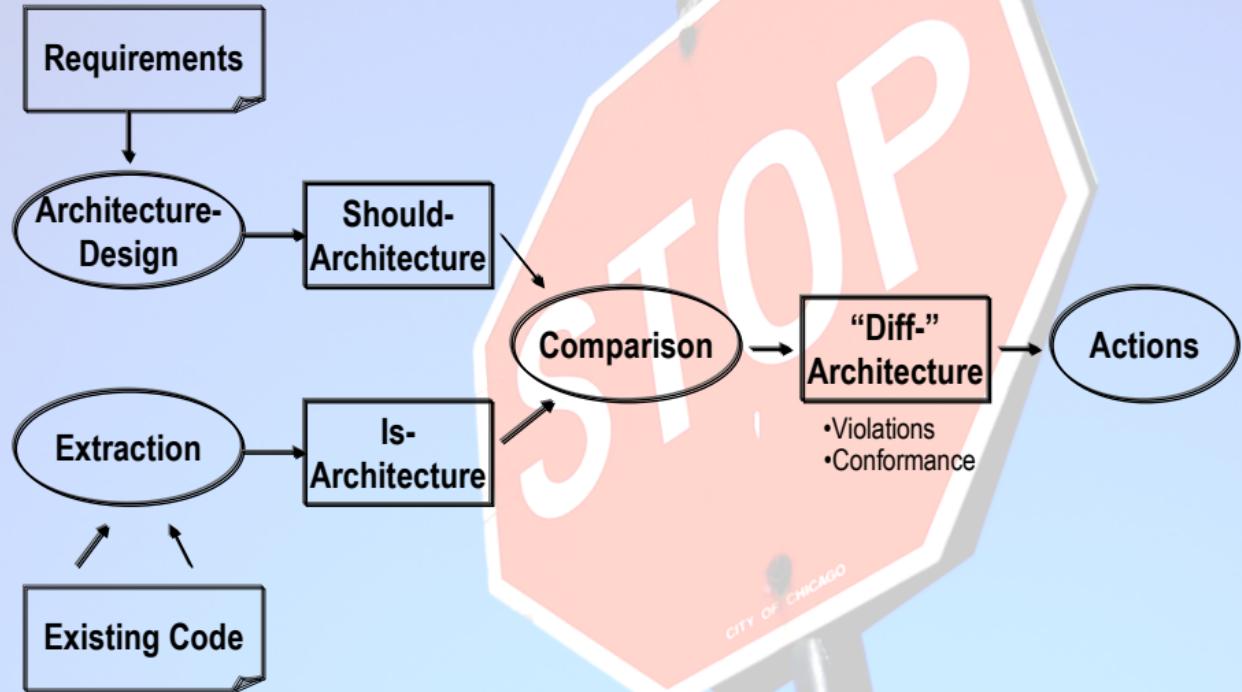
Displaying results

Monitoring changes, trends

Rating of Architecture

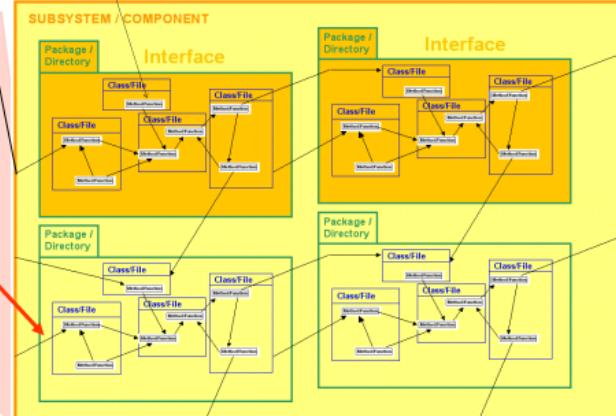
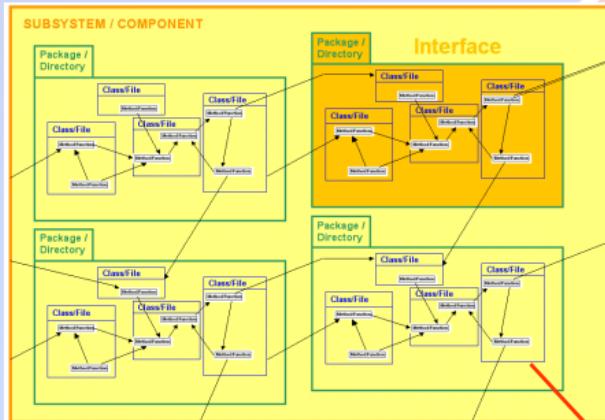
→ Requirements for Architectural Analysis Tools

# Architecture Analysis (deviation)



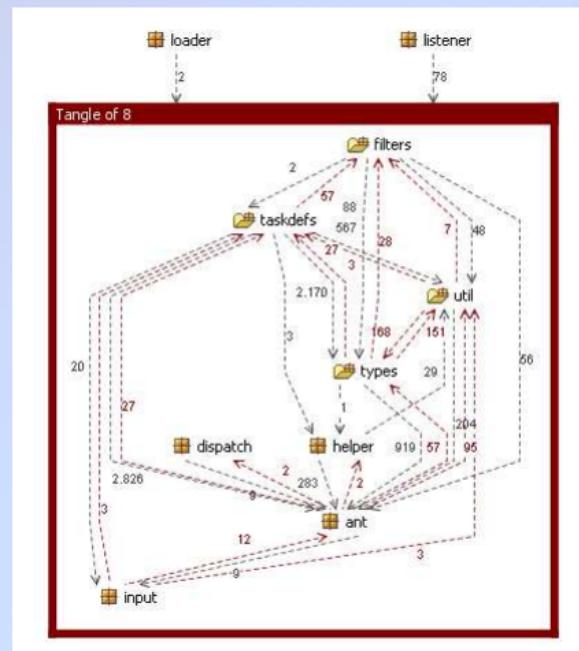
# Drill-Down + Aggregation:

Component → System → Package → Class → Src-Line



# Displaying results:

## Graphical and/or numerical



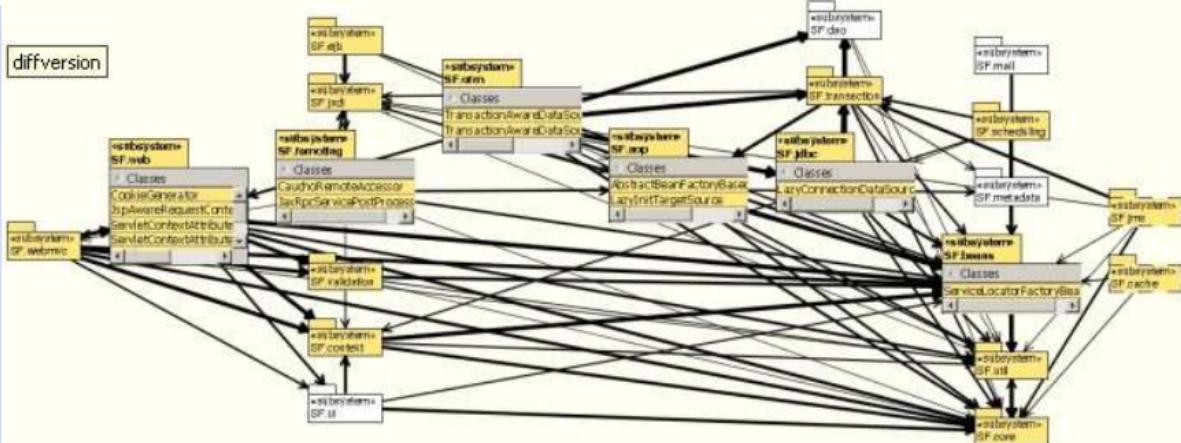
# Monitoring changes, trends (QA)

Violations of Architecture Model 'Overview' for v1 = spring20041212\_113 and v2 = spring20050131\_114

ID	subRefing	ID	subRefed	errorKind	v1	v2	diff	changes
SU	SF.transaction	SU	SF.jndi	UPWARD	10	10	0	
SU	SF.transaction	SU	SF.acp	UPWARD	57	60	3	x
SU	SF.remoting	SU	SF.webmvc	UPWARD	3	3	0	
SU	SF.remoting	SU	SF.web	UPWARD	4	4	0	
SU	SF.orm	SU	SF.web	UPWARD	25	25	0	
SU	SF.jdbc	SU	SF.jndi	UPWARD	5	5	0	
SU	SF.web	SU	SF.webmvc	UPWARD	10	10	0	
SU	SF.scheduling	SU	SF.transaction	UPWARD	19	19	0	
SU	SF.scheduling	SU	SF.jdbc	UPWARD	2	2	0	
SU	SF.scheduling	SU	SF.context	UPWARD	9	9	0	
SU	SF.beans	SU	SF.jndi	UPWARD	3	3	0	

ion etc.

diffversion

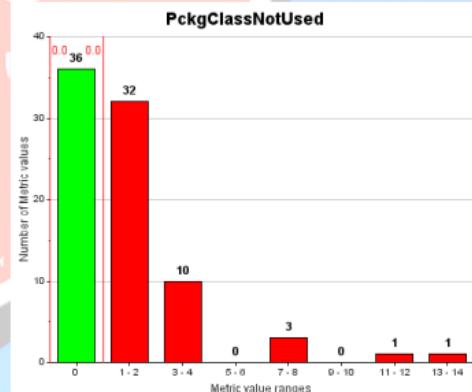
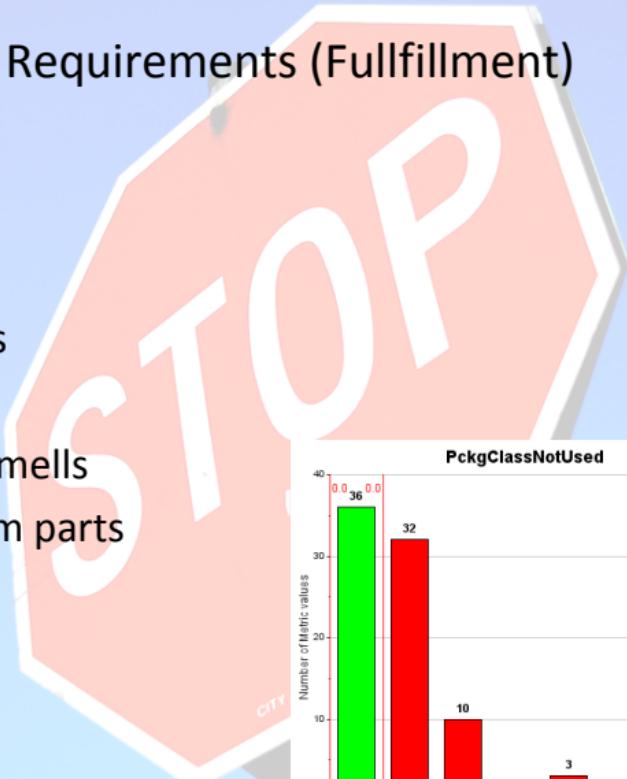


# Rating of Architecture

NO Rating of external Requirements (Fullfillment)

Internal Quality:

- Cycles
- Coupling of Modules
- Stability of API
- Anti-Patterns, Bad Smells
- Reuse/Extract system parts



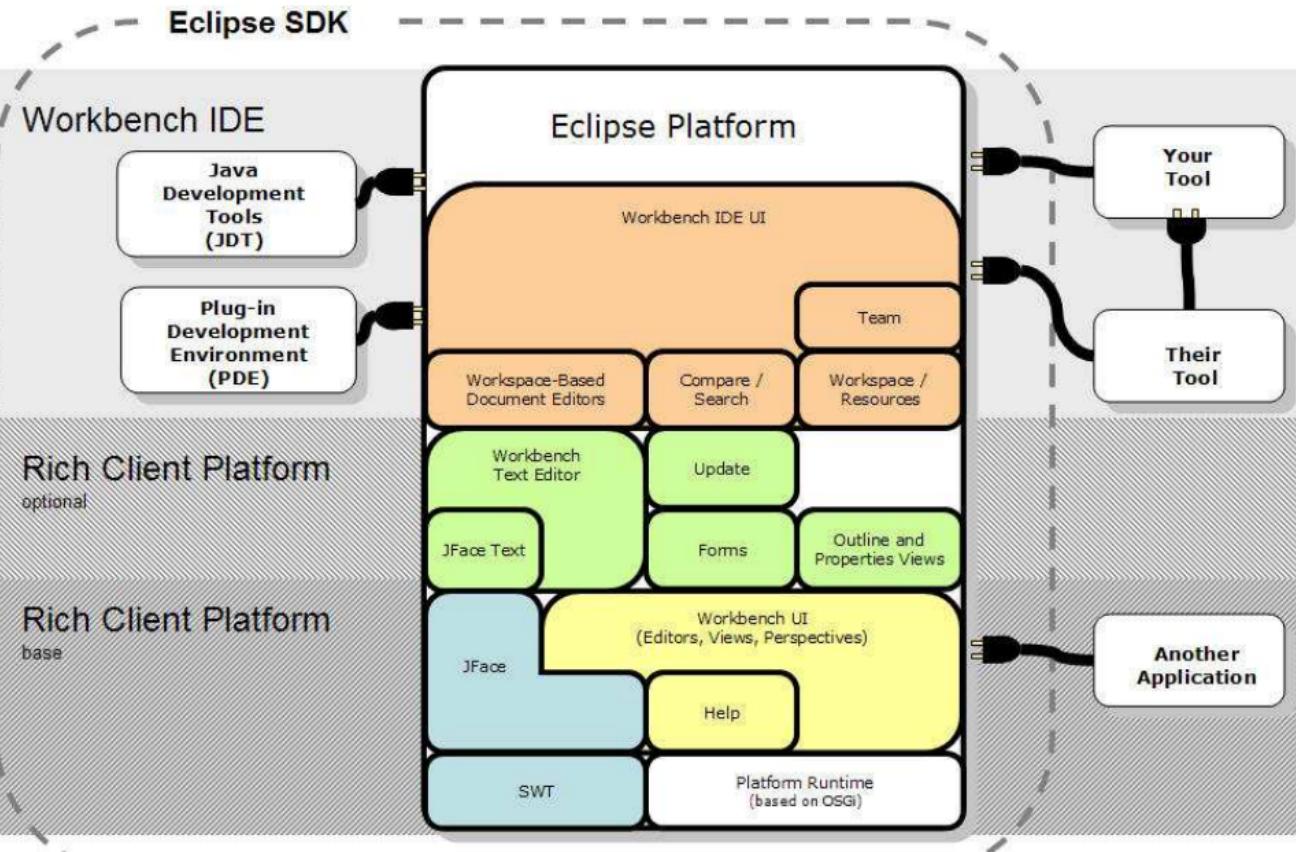
# Eclipse: Architectural Analysis

-JDT

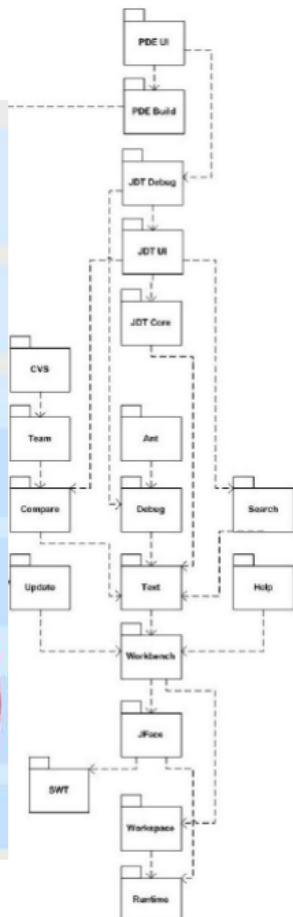
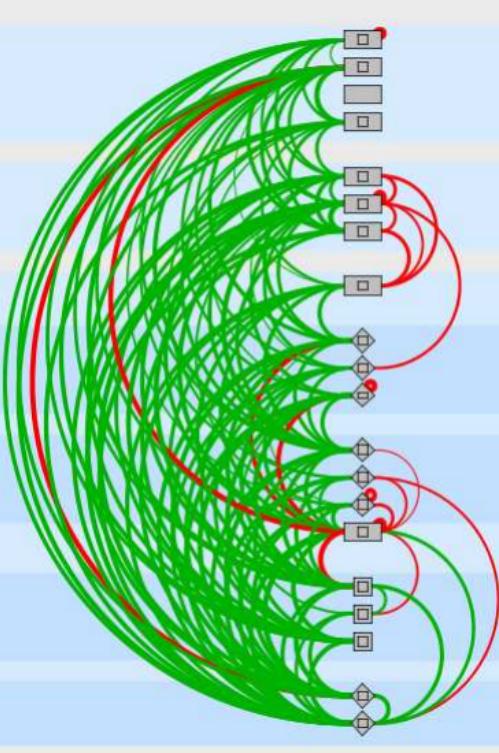
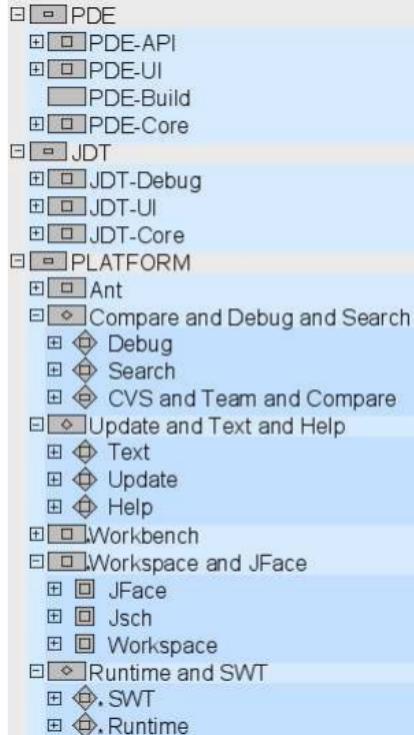
eclipse

-CDT

# Eclipse Architecture

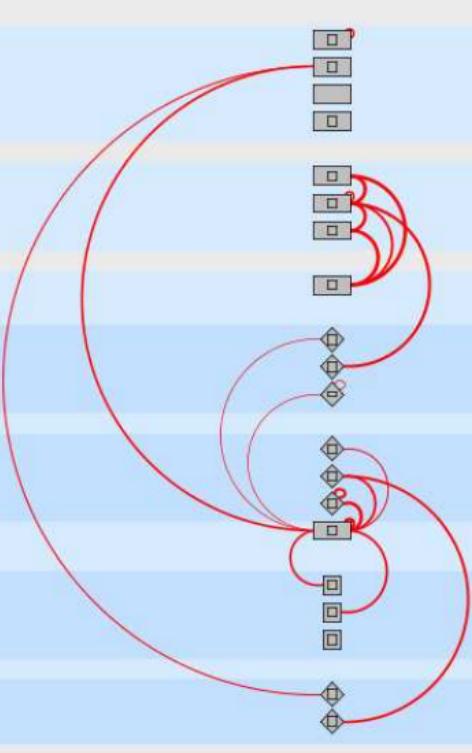


# Eclipse Architecture

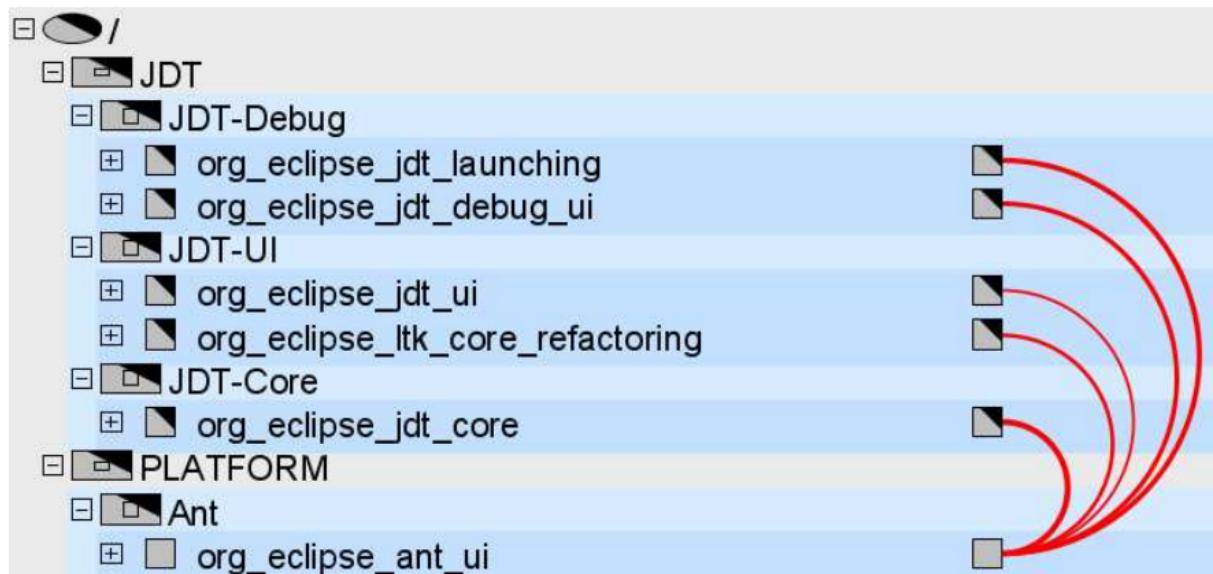


# Eclipse Architecture

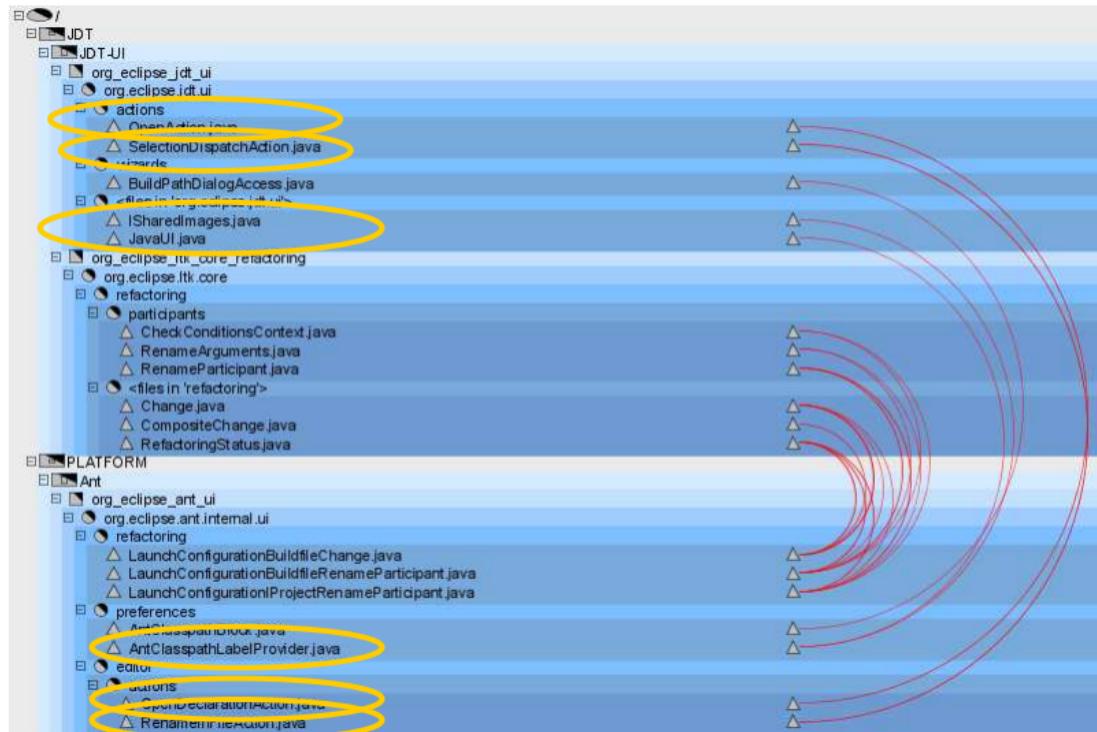
■	└ PDE
└	└ PDE-API
└	└ PDE-UI
└	└ PDE-Build
└	└ PDE-Core
■	└ JDT
└	└ JDT-Debug
└	└ JDT-UI
└	└ JDT-Core
■	└ PLATFORM
└	└ Ant
└	└ Compare and Debug and Search
└	└ Debug
└	└ Search
└	└ CVS and Team and Compare
└	└ Update and Text and Help
└	└ Text
└	└ Update
└	└ Help
└	└ Workbench
└	└ Workspace and JFace
└	└ JFace
└	└ Jsch
└	└ Workspace
└	└ Runtime and SWT
└	└ SWT
└	└ Runtime



## E3.4: Plattf:Ant → JDT:\*



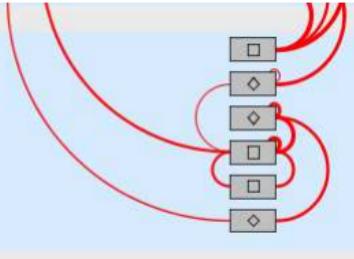
# E3.4: Plattf:Ant → JDT:UI



# Plattform: CVS → Workb (internal)

## PLATFOR

- +  Ant
- +  Compare and Debug and Search
- +  Update and Text and Help
- +  Workbench
- +  Workspace and JFace
- +  Runtime and SWT



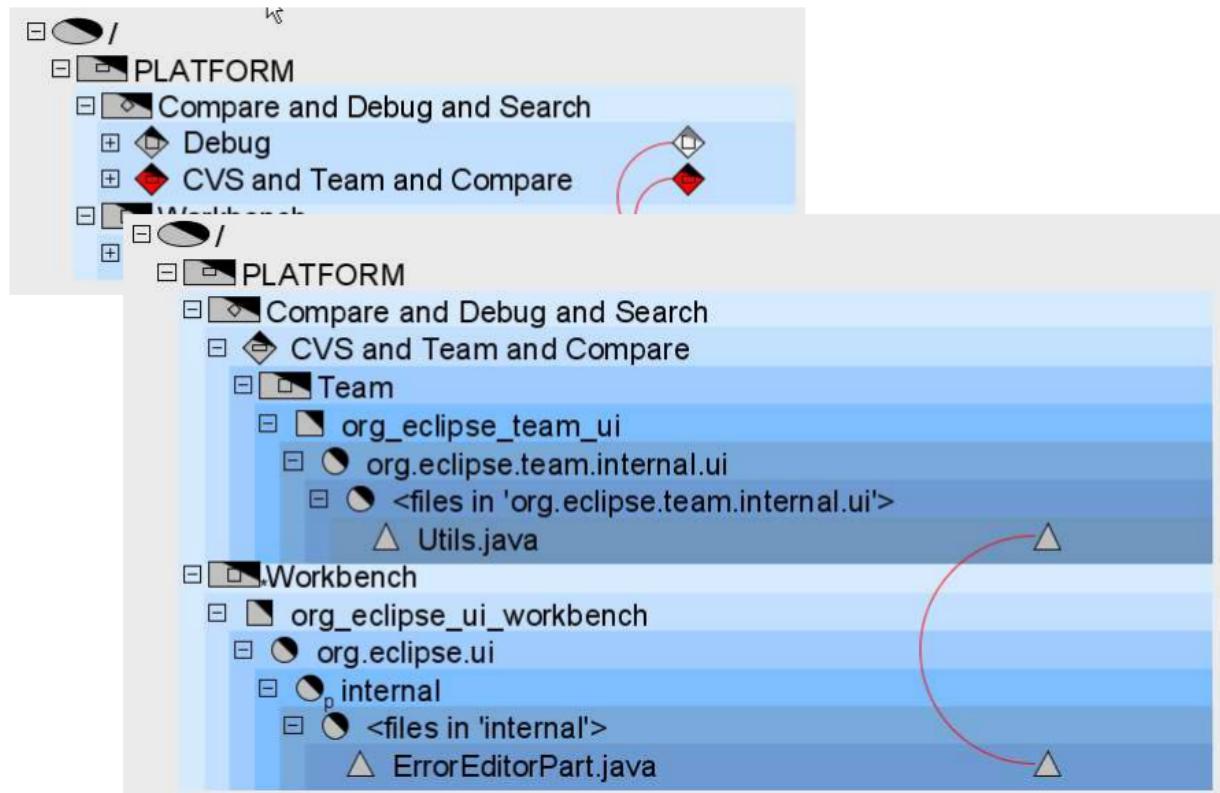
## /

## PLATFOR

- Compare and Debug and Search
  - +  Debug
  - +  CVS and Team and Compare
- Workbench
  - +  org\_eclipse\_ui\_workbench



# Plattform: CVS → Workb (internal)



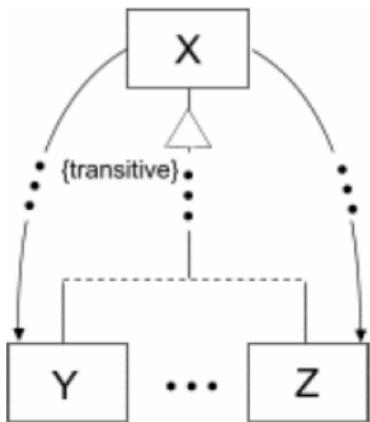
# Team-UI → UI-workbench (internal)

[https://bugs.eclipse.org/bugs/show\\_bug.cgi?id=90582](https://bugs.eclipse.org/bugs/show_bug.cgi?id=90582)

E:\e3.4cdt5\ eclipse-SDK-  
1031  
1032     public st...  
1033         String ...  
1034         try {  
1035             IEd...  
1036             //  
1037             if  
1038             ->  
1039  
1040         }  
1041         if  
1042         }  
1043         ret...  
1044     } catch...  
1045         if  
1046         }  
1047         } e...  
1048     }  
1049     }  
1050     }  
1051     }  
1052 }  
  
Bug 90582 – [EditorMgmt] (regression) openEditor does not throw PartInitException anymore - Mozilla Firefox  
File Edit View History Bookmarks ScrapBook Tools Help  
eclipse.org https://bugs.eclipse.org/bugs/show\_bug.cgi?id=90582  
Bug 90582 – [EditorMgmt] (regression) openEditor does not throw PartInitException anymore - Mozilla Firefox  
be to get this fixed for 3.5? I am willing to upgrade to P2 again based on your feedback.  
Michael Valenta 2008-11-17 10:49:21 EST Comment 46  
We just got hit by this again (i.e. bug report from customer) since we did not implement the CVS hack workaround in Jazz because it requires the use of workbench internals. If the internal editor fails to open on an IStorageInput, we need a fallback which allows the default text editor to be opened. Perhaps the ErrorEditorPart could just have a button that provides the option to open the file in a text editor.  
Here's the hack CVS does to get around this problem:  
plugin: org.eclipse.team.ui  
class: org.eclipse.team.internal.ui.Utils  
method:  
    public static IEditorPart openEditor(IWorkbenchPage page,  
    FileRevisionEditorInput editorInput) throws PartInitException {  
        String id = getEditorId(editorInput);  
        try {  
            IEditorPart part = page.openEditor(editorInput, id);  
        }

# Dependent BaseClass

- Type:
  - Design Problem
- Problem:
  - one or more Methods shall implement different behavior, depending on the type, passed in
- Context:
  - make “extensible” systems, frameworks
- Forces:
  - Programming languages offer, instanceof/typeid funcs.
- Antipattern:
  - Methods of the baseclass, depend on derived classes, e.g. accessing their members, doing switch/case depending on type information



# AntiPatterns / Bad Smells:

Metrics/1000 Classes	Eclipse	JDK
Dependent Baseclass:	1	16
Multiple Interface Inher.	4	18
Abstractable Methods	80	60
Abstractable Attributes	45	170
Unused Classes	50	150
Unused Methods	950	2500
Unused Attributes	30	20

# Code Clones

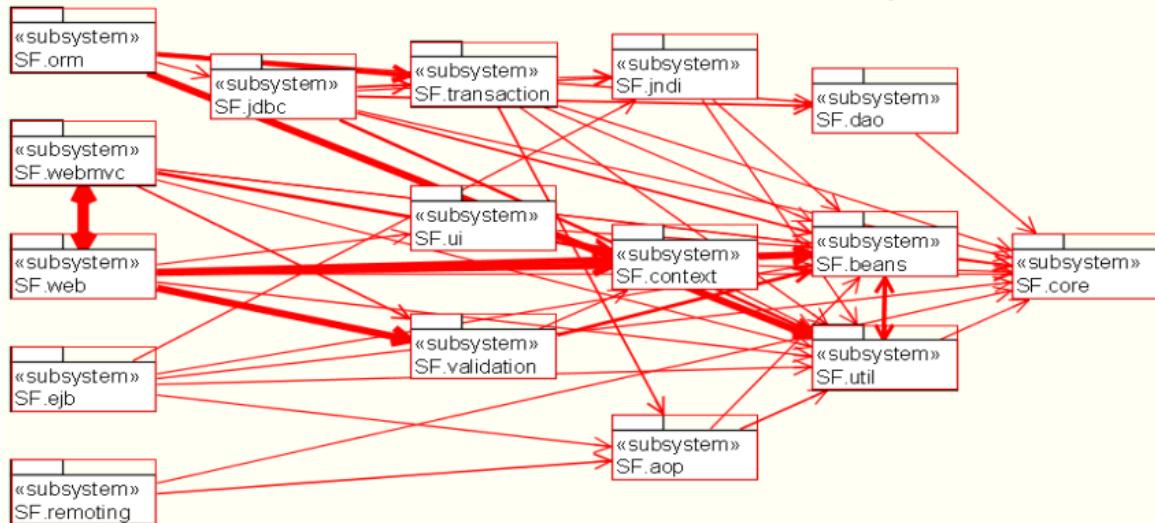
- identical Files
  - E2.0 JDT,CDT
    - jdt\debug\internal\ui\dialogfields\ListDialogField.java
    - cdt\debug\internal\ui\dialogfields\ListDialogField.java
  - E3.4 CDT: identical packages
    - cdt\debug\internal\ui\dialogfields,
    - cdt\debug\mi\internal\ui\dialogfields
- variation of former identical Files
  - E3.4 JDT,CDT
    - jdt\debug\internal\ui\dialogfields\ListDialogField.java
    - cdt\debug\internal\ui\dialogfields\ListDialogField.java

# Discover Architecture:

## Call-Relationships

Using...

Used by...



- Business Modules
- Public, exported Methods

- Utility Modules, Infrastructure
- Private, internal Methods

# Pros/Cons Architecture Analysis Tools

Wood/Trees !

Internal Structure !

Metrics !

Internal Quality !

Clones !

Multiple Languages

Monitoring Changes

Virtual Refactoring

API Design ?

Additional aspects ?

Metrics ?

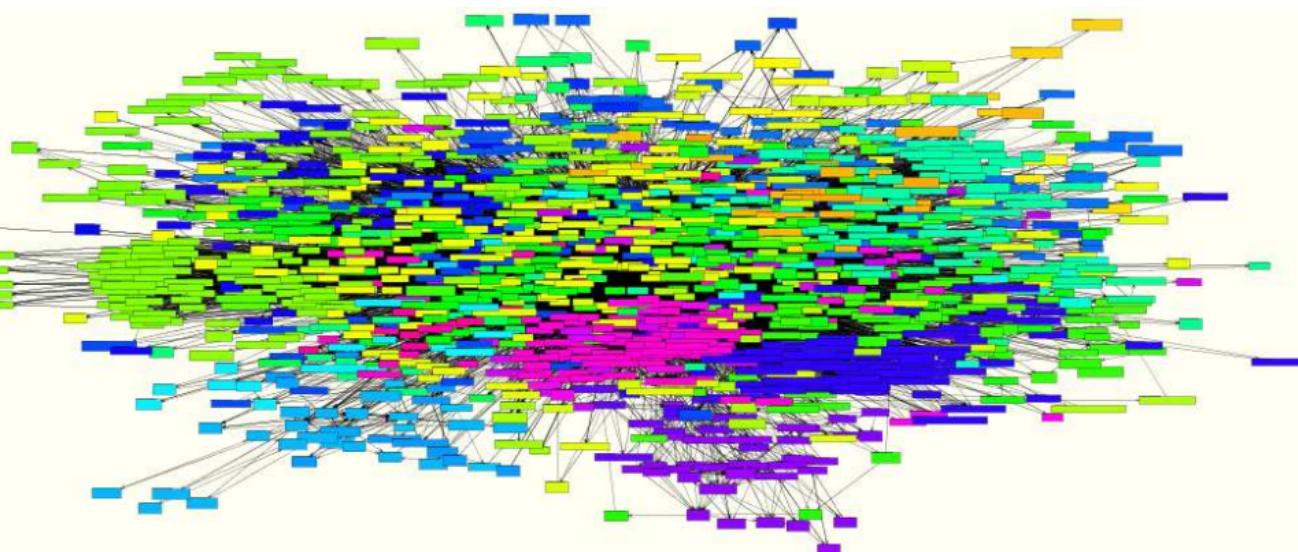
Functional Redundancy?

# Rating of Architecture

1315 classes in 229 packages all depend on each other !!!

classes.zip, rt.jar (BIG BALL OF ... ;-)

JDK^H^Hdecay



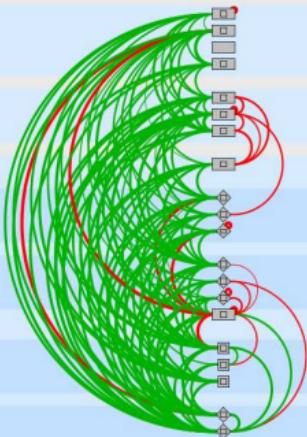
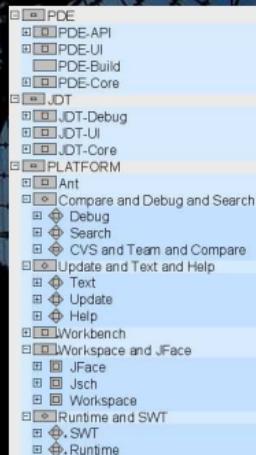
# Still manageable?



# Rating Eclipse Architecture

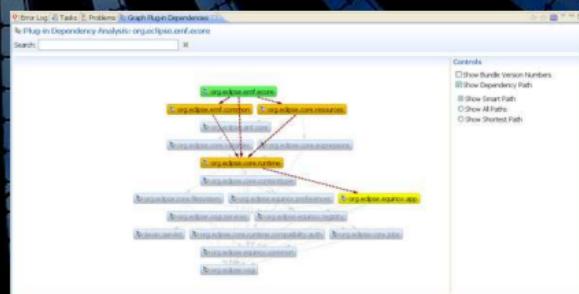
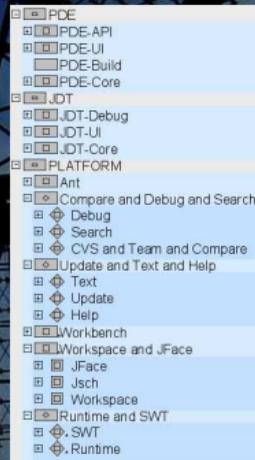
Erosion not really dramatic, but happened/s

- large codebase,  
structured in plugins
- OSGI helps a lot
- API Police, Guidelines  
upcoming PDE tools



# Rating Eclipse Architecture:

- Violations
    - Upper layers
    - Internal access
  - Small amount of
    - AntiPatterns
  - Some amount of
    - CodeClones
  - OSGI is not enough
- People AND Tools !



# URLs, References:

<http://se-radio.net/podcast/2008-10/episode-115-architecture-analysis>

Thanks to:

- Headways: Structure101:
  - Ian Sutton
  - Paul Hickey
- hello2morrow: SotoArc
  - Thomas Schoen
  - Norman Wenzel
  - Heinrich Rust

# Pictures under CC license from flickr:

1191285966\_d701fcb1c3\_b\_flickr\_bluemeat.jpg

3425532267\_bd74526b23\_b\_flickr\_ttwo.jpg

1813471845\_5a4be999dc\_o\_flickr\_thatcanadiangrrl.jpg

2401122677\_c25dea1233\_b\_flickr\_EnglishGirlAbroad.jpg

501709581\_f3729ceaebe\_b\_flickr.jpg

# **Stop the Software Architecture Erosion**



**Bernhard Merkle**

**Central Research & Development**

**Software-Engineering**

**SICK-AG Waldkirch**

**[mailto: Bernhard.Merkle@gmail.com](mailto:Bernhard.Merkle@gmail.com)**

**Contact on [linkedin.com](#) or [xing.com](#)**

