

Dr. Michael Eichberg
Software Engineering
Department of Computer Science
Technische Universität Darmstadt

Software Engineering

Software Quality



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Software Quality

- **Software Quality Factors**

B. Meyer; Object-oriented software construction; Prentice Hall, 1997



TECHNISCHE
UNIVERSITÄT
DARMSTADT

We distinguish between **internal** and **external** **software quality factors.**

Software Quality | 3

- The internal quality factors can only be perceived by computer professionals
- The external quality factors are ultimately the relevant ones, as they are perceived by the user

However, the external quality factors depend on the internal quality factors.

We distinguish between **internal** and **external** **software quality factors.**

Software Quality - Major External Software Quality Factors | 4

- Correctness
- Robustness
- Extendibility
- Reusability
- Compatibility
- Efficiency
- Portability
- Ease of use
- Functionality

The user
encompasses all
stake holders:
- the owner,
- the "end user",
- the administrator,
- ...

Correctness is the ability of software products to perform their tasks as defined by their specification.

Major External Quality Factors | 5

- To achieve correctness a precise requirements definition is needed
- Correctness is usually only conditional - we guarantee the correctness of our program on the assumption that the lower layers - upon which our product is built - are correct
(E.g. we assume that a processor calculates correctly, that the compiler compiles our program correctly, ...)

Robustness is the ability of software systems to react appropriately to abnormal conditions.

Major External Quality Factors | 6

- Robustness characterizes what happens “outside of the specification”
- Robustness complements correctness

Extendibility characterizes the ease of adapting software products to changes of the specification.

Major External Quality Factors | 7

- Important principles to achieve extendibility:
 - Design simplicity
A simple architecture is easier to adapt.
 - Decentralization
Autonomous modules (modules which have minimal coupling to other modules → Software Engineering Design & Construction) are easier to change.

Change is pervasive in software development.

- **Reusability** is the ability of software elements to serve for the construction of many different applications
- **Compatibility** is the ease of combining software elements with others
- **Portability** characterizes the ease of transferring software products to various hardware and software environments (i.e., porting it from Android to iOS; porting it from Windows to Linux,...)

Efficiency is the ability of a software system to place as few demands as possible on hardware resources.

Major External Quality Factors | 9

- Resources are the processor time, the space occupied in internal and external memories, the bandwidth used in communication devices, ...
...
- Always try to use “good” algorithms over “bad” ones, because a computer that is twice as fast as a previous model can handle problem sizes near 2^*N if the algorithm’s complexity is $O(n)$.
Do ask yourself: If the complexity is $O(2^n)$ a computer that is twice as fast can handle problems of size?

Do not worry how fast it is unless it is also right! Efficiency nearly always have to be balanced with other goals.

Functionality

characterizes the extent of possibilities provided by a system.

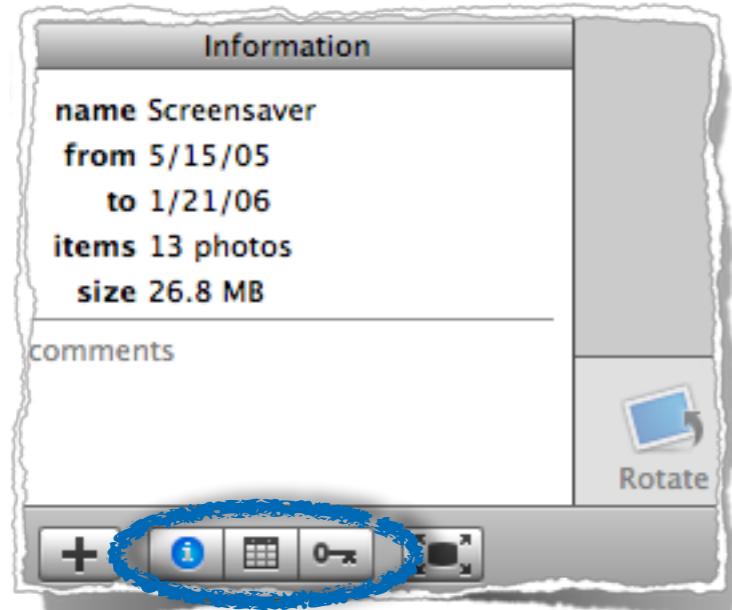
Major External Quality Factors | 10

- Avoid featurism; remain consistent with existing features if you add new ones

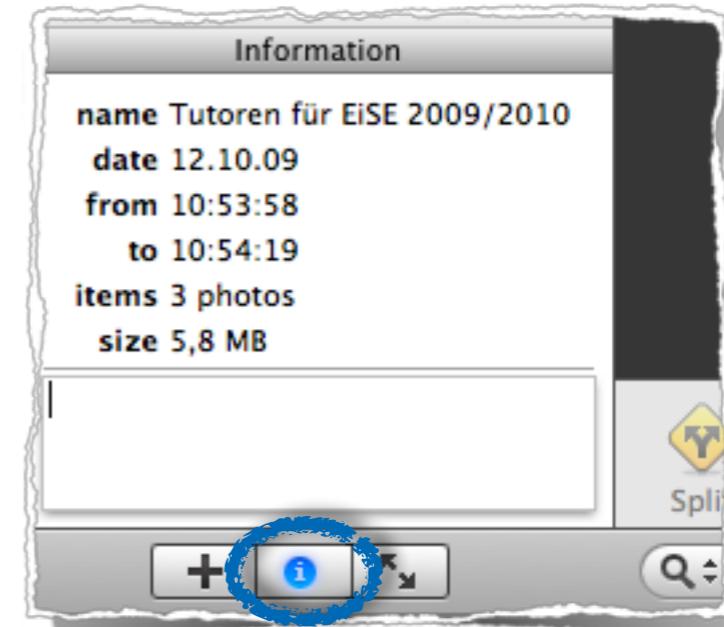
Ease of Use

is the ease with which people of various backgrounds and qualifications can learn to use software products and apply them to solve problems.

Major External Quality Factors | 11



iPhoto '06



iPhoto '09

Software Quality

- ... or the lack thereof.



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Software failures can be disastrous.

- Therac-25

People died due to an overdose of radiation (1985)
- Ariane 5

A system from Ariane 4 was reused but the specification was ignored (1996)
- Mars Climate Orbiter

There was some confusion about the units (i.e. metric system or english system) that are used (1999).
- ...

Software failures can be disastrous.

- hessische Schulsoftware LUSD

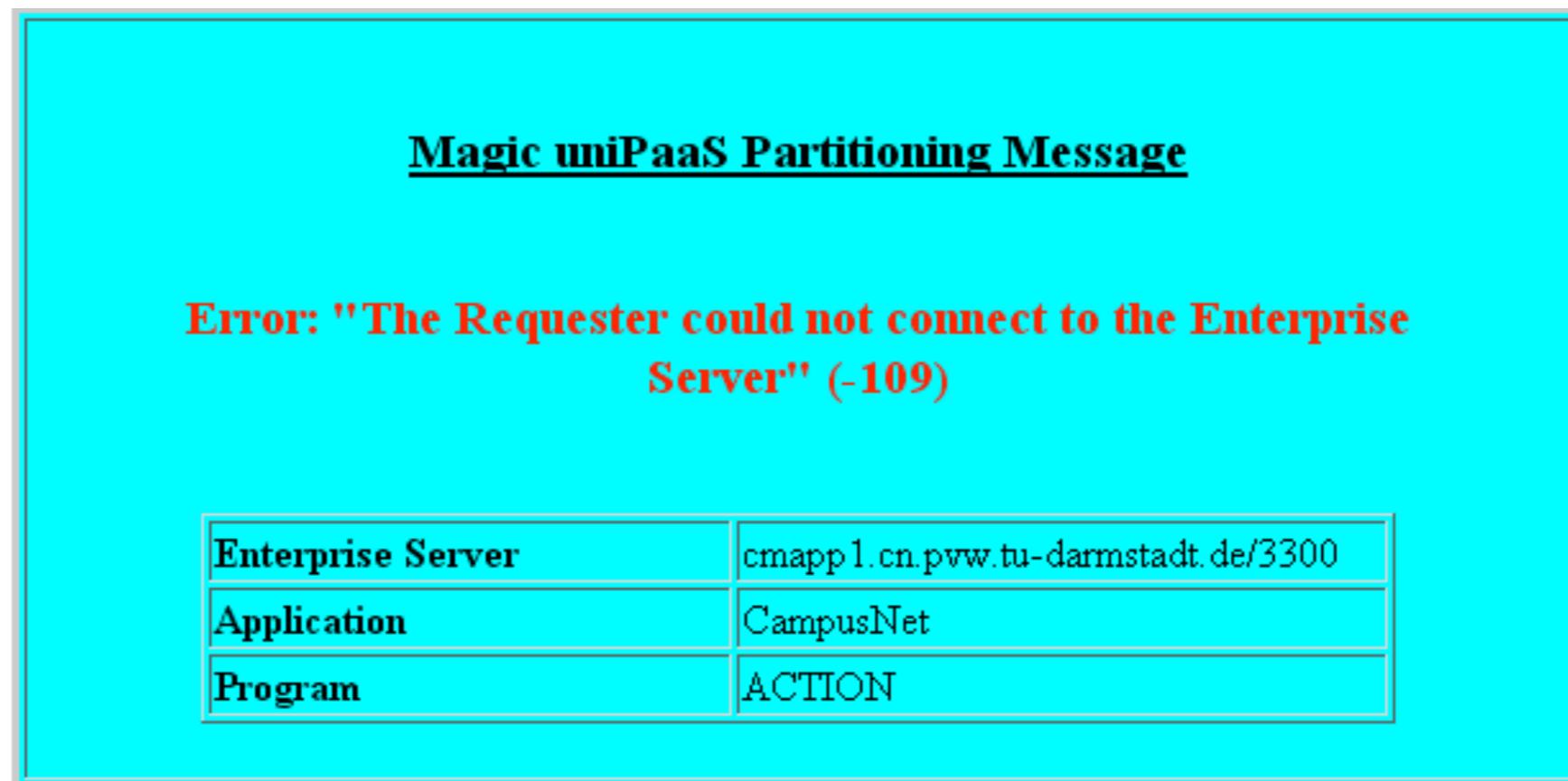
"just" unusable (2007)

• ...

The screenshot shows a news article from the website **TEC CHANNEL**. The header includes the logo "TEC CHANNEL IT EXPERTS INSIDE", a search bar labeled "Suchbegriff hier", and a sorting dropdown. The main navigation menu includes links for **HOME**, **FOKUS**, **WHITEPAPER**, **SICHERHEIT**, **NETZWERK**, **SERVER**, and **ST...
SERVICE**. Below the menu are links for **Aktuelle Themen**, **News**, **News-Archiv**, **Newsletter**, **PDF Newsletter**, and **PDA-News**. The article title is **THEMENÜBERBLICK / NEWS** followed by **Weitere News**. The main headline is **Hessen erlebt Desaster mit neuer Schulsoftware von CSC**, dated **Vom 14.09.2007**. The text below the headline states: **Das Kultusministerium hat eine für 20 Millionen Euro entwickelte Verwaltungssoftware an den Schulen installieren lassen, die nicht funktioniert.** A small image shows people standing outside a building, likely a school. The bottom text provides more context: **Bereits seit dem vergangenen Schuljahr versuchen rund 2000 hessische Schulen mit der neuen Schulverwaltungssoftware LUSD (Lehrer- und Schülerdatenbank) zu arbeiten. Bis heute ist sie jedoch unbrauchbar. Entwickelt wurde die Schulsoftware von CSC. Start der Konzeption und der Entwicklung war der 1. Juni 2006. Mit der Implementierung in den Schulen hatte CSC im Oktober 2006 begonnen.**

Lack of software quality.

- CampusNet error message shown to the end user (2010)



Arbeitsablauf Aktion (Einfache Freigabe)

Allgemein Formular Historie

Bearbeiter

Priorität mittel Termin

Kommentar

Letzter Arbeitsschritt

Bearbeiter michael-eichberg

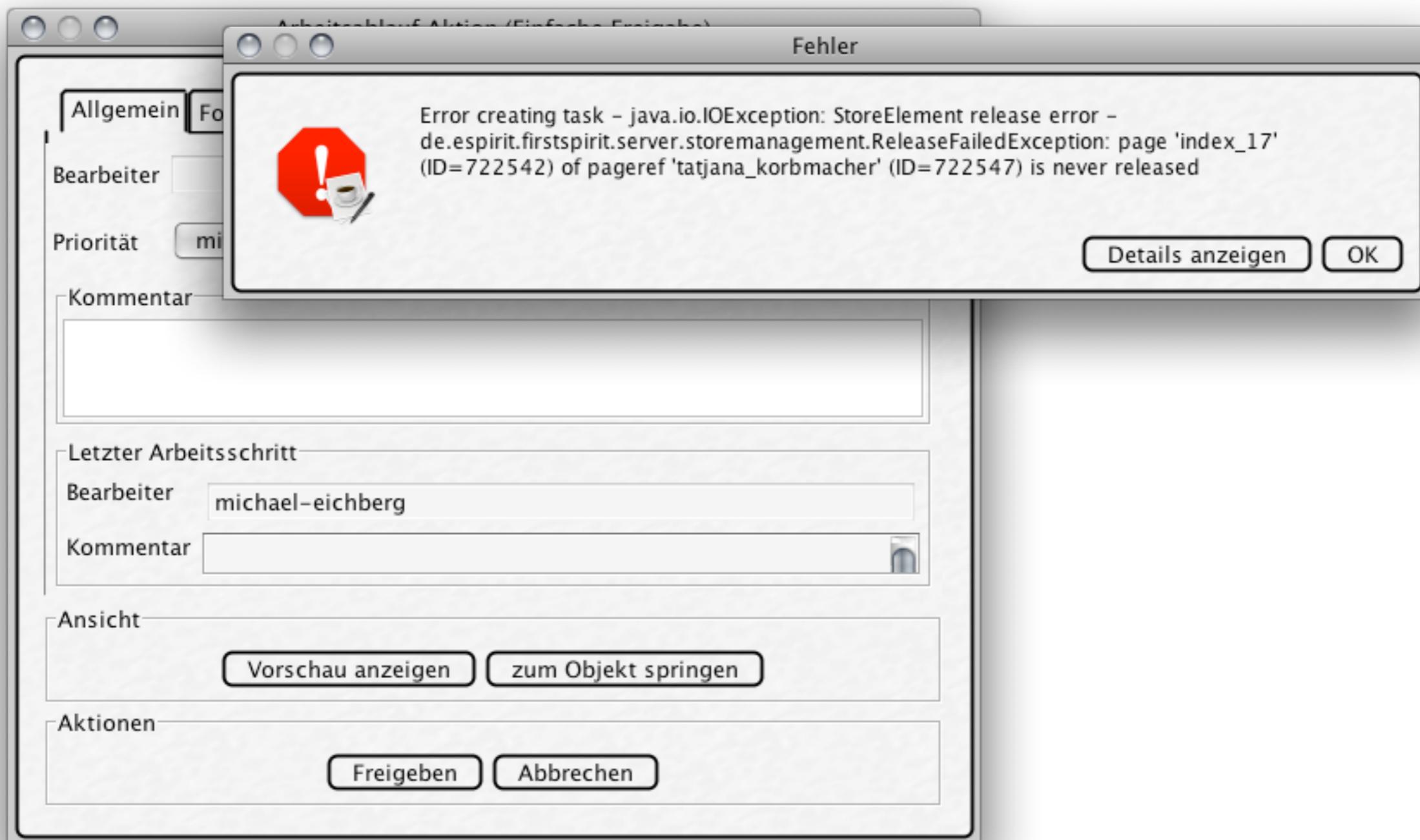
Kommentar

Ansicht

Vorschau anzeigen zum Objekt springen

Aktionen

Freigeben Abbrechen



Arbeitsblatt-Aktion (Einfache Funktion)

Fehler

Allgemein

Bearbeiter

Priorität

Kommentar

Letzter Arbeitsschritt

Bearbeiter michael-eichberg

Kommentar

Ansicht

Vorschau anzeigen

Aktionen

Freie

Error creating task – java.io.IOException: StoreElement release error –
de.espirit.firstspirit.server.storemanagement.ReleaseFailedException: page 'index_17'
(ID=722542) of pageref 'tatjana_korbmacher' (ID=722547) is never released

Details anzeigen OK

Fehler

Error creating task – java.io.IOException: StoreElement release error –
de.espirit.firstspirit.server.storemanagement.ReleaseFailedException: page 'index_3' (ID=722534)
of pageref 'roman_knoell' (ID=722541) is never released

java.io.IOException: StoreElement release error –
de.espirit.firstspirit.server.storemanagement.ReleaseFailedException: page 'index_3' (ID=722534)
of pageref 'roman_knoell' (ID=722541) is never released

at
de.espirit.firstspirit.server.taskmanagement.TaskImpl.doTransition(TaskImpl.java:988)
at
de.espirit.firstspirit.server.taskmanagement.TaskImpl.doTransition(TaskImpl.java:811)
at
de.espirit.firstspirit.client.action.WorkflowAction.startWorkflow(WorkflowAction.java:446)
at de.espirit.firstspirit.client.action.WorkflowAction.startWorkflow(WorkflowAction.java:74)
at de.espirit.firstspirit.client.action.WorkflowAction.startWorkflow(WorkflowAction.java:80)
at
de.espirit.firstspirit.client.gui.workflow.WorkflowPopup\$WFStartAction.actionPerformed(WorkflowPopu
p.java:413)
at javax.swing.AbstractButton.fireActionPerformed(AbstractButton.java:2028)
at javax.swing.AbstractButton\$Handler.actionPerformed(AbstractButton.java:2351)
at javax.swing.DefaultButtonModel.fireActionPerformed(DefaultButtonModel.java:387)
at javax.swing.DefaultButtonModel.setPressed(DefaultButtonModel.java:242)
at javax.swing.AbstractButton.doClick(AbstractButton.java:389)
at javax.swing.plaf.basic.BasicMenuItemUI.doClick(BasicMenuItemUI.java:1220)
at apple.laf.CUIAquaMenuItem.doClick(CUIAquaMenuItem.java:119)
at
javax.swing.plaf.basic.BasicMenuItemUI\$Handler.mouseReleased(BasicMenuItemUI.java:1261)

Details verbergen OK

Missing software quality in commercial software.

- Lufthansa Buchungssystem
“Totalausfall” (2004)

• ...

Handelsblatt.com

Konsequenz aus System-Ausfall

14.10.2004

Lufthansa will Check-In-Technik besser absichern

Nachdem das weltweiten Check-In-Systems bei der Lufthansa vor drei Wochen komplett ausgefallen ist zieht die Airline nun Konsequenzen.

Eine Entscheidung darüber werden die Lufthansa-Manager am Donnerstag in Frankfurt-

HB FRANKFURT. Der für die Technik zuständige Managerin Barbara Franke erläuterte,

bei dem Ausfall das vorhandene Risiko nicht abgewältigt zu haben.

Entscheidender Stelle auf das ausfallende System umgestellt. Am 23. September weltweit die Flugreisen eingekauft und eingeccheckt. Neben zahlreichen Vier-Sterne-Hotels Prinzipiell sei ein eigenständiges Back-up-System liege beim Kunden.

Nach Einschätzung von Technikchef Barbara Franke sei der Betrieb bereit stünde, binnen wenigen Minuten wieder auf das System zurückzuschalten.

Der System würde rund zehn Mill. € kosten.

Köln bei Technikanbietern für den Austausch.

Über möglichen Schadenersatz besteht bei Lufthansa noch keine Klarheit.

„Ob und in welcher Höhe Schadenersatzforderungen geltend gemacht werden, wird momentan noch geprüft“, teilte die Unternehmens-Sprecherin mit.

Eine Wiederholung der konkreten Ursache für den Check-In-Ausfall gilt mittlerweile als ausgeschlossen.

„Es handelte sich um eine Verkettung von gleich drei Problemen“, sagte Systems-Chef

Franke. Die US-Firma Unisys, deren Betriebssystem Lufthansa beim Check-In verwendet, hatte in der Nacht ein Software-Update ausgeführt. Im Gefolge davon kam es zu einem Systemabsturz, weil eine Speicherdatei vollgelaufen war.

... hatte in der Nacht ein Software-Update ausgeführt. Im Gefolge davon kam es zu einem Systemabsturz, weil eine Speicherdatei vollgelaufen war....

- Lufthansa Buchungssystem
“Totalausfall” (2009)

sueddeutsche.de

• ...

Computerpanne bei Lufthansa

30.09.2009, 12:26

Mit Zettel und Stift musste die Lufthansa heute ihre Passagiere einchecken. Eine Computerpanne hatte den Check-In lahmgelegt.

Mit Verspätungen muss wegen der Computerpanne noch bis morgen gerechnet werden. (Foto: ddp)

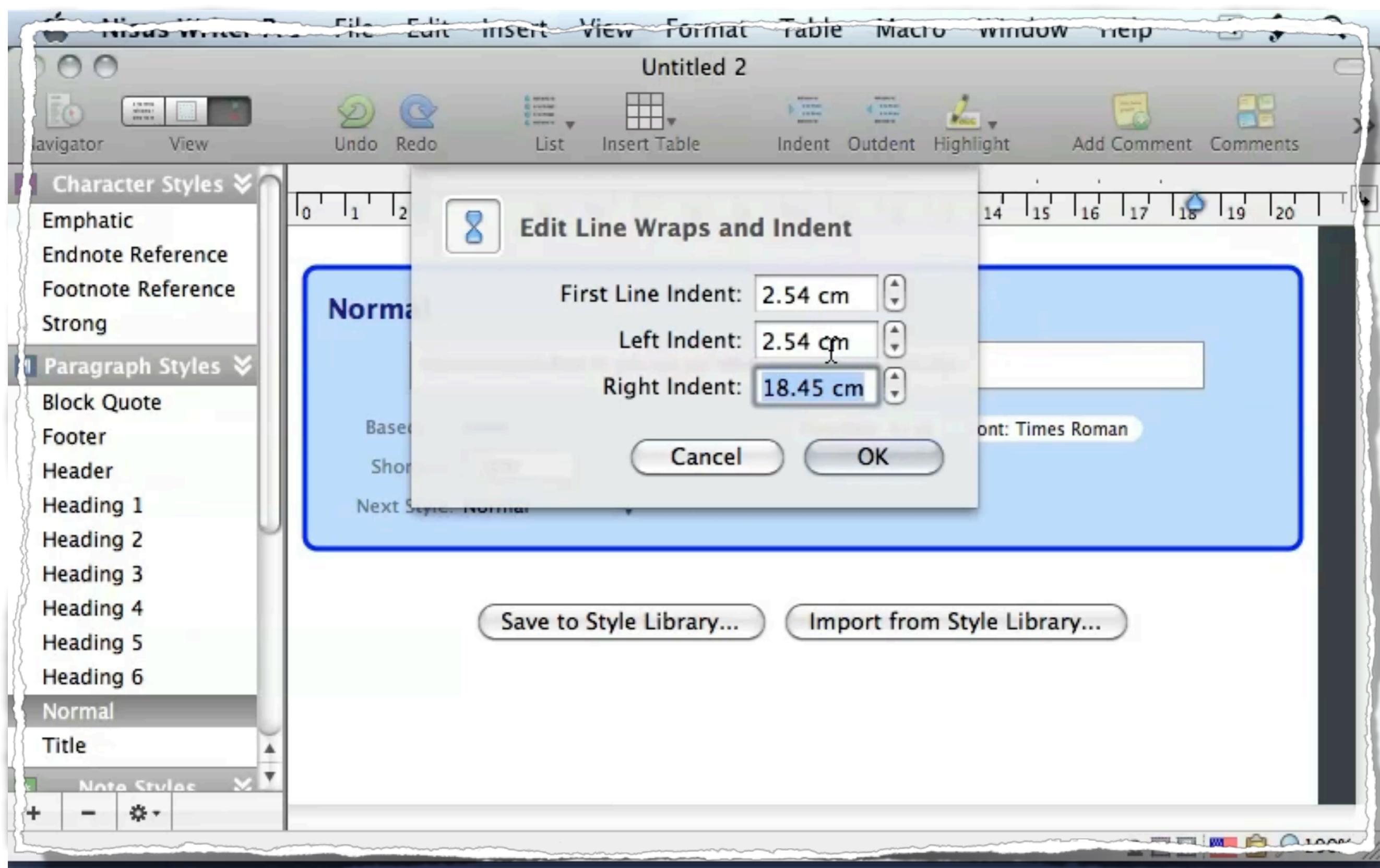
Ein Ausfall des zentralen Lufthansa-Check-In-Systems hat weltweit zu Verzögerungen bei der Abfertigung sowie zu Verspätungen und einzelnen Flugausfällen geführt.

Das System kam kurz vor 04.00 Uhr während eines routinemäßigen Software-Updates zum Stillstand, wie ein Sprecher sagte. Zwar habe man den Server um 08.00 Uhr wieder starten können. Bis zum Mittwochabend könne es aber zu Verspätungen kommen.

Wegen des Systemausfalls musste die Lufthansa weltweit auf manuelles Einchecken umstellen. Passagiere wurden per Hand mit Stift und Papier eingekennigt werden, sagte Lufthansa-Sprecher[...]

Missing software quality in commercial software.

Nisus Writer Pro 1.4 - 2010 | 21



Missing software quality in commercial software.

Duden Office Bibliothek - 2010 | 22

The screenshot shows the 'Office-Bibliothek Express' application window. The menu bar includes 'Office', 'Bibliothek', 'Ablage', 'Bearbeiten', 'Suchen', 'Fenster', 'Extras', and 'Hilfe'. The title bar says 'Office-Bibliothek Express'. A toolbar has buttons for 'Suche' (Search), a magnifying glass icon, and file operations. Below the toolbar, there's a search field labeled 'Suche:' with a dropdown menu showing 'Gesamt' (General) and 'Suchergebnis' (Search results). A link 'Erweiterte Suche' (Advanced Search) is also present. On the left, a sidebar lists several books with yellow book icons and labels like '404', 'A', 'a', 'a', 'A', 'à', 'Installierte Bücher' (Installed Books), and 'DUDEN Die deutsche Rechtschreibung'. A central panel displays the search results for the word 'about'. The results are numbered 1 through 6, with definitions in German and English. The first definition is: 'a·bout [ə'baʊt] I. prp. 1. um, um... herum; 2. umher in (dat.): **wander about the streets**; 3. bei, auf (dat.), an (dat.), um, in (dat.): (**somewhere**) **about the house** irgendwo im Haus; **have you any money about you?** haben Sie Geld bei sich?; **look about you!** sieh dich um!; **there is nothing special about him** an ihm ist nichts Besonderes; 4. wegen, über (acc.), um (acc.), von: **talk about business** über Geschäfte sprechen; **I'll see about it** ich werde danach sehen od. mich darum kümmern; **what is it about?** worum handelt es sich?; 5. im Begriff, da bei: **he was about to go out**; 6. beschäftigt mit: **what is he about?** was macht er (da)?; **he knows what he is about** er weiß, was er tut'.

Suche:

Gesamt Suchergebnis Erweiterte Suche

404
A
a
a
A
à

Installierte Bücher

DUDEN Die deutsche Rechtschreibung

Vordefinierte Buchauswahl

Alle Bücher ausgewählt ...

a·bout [ə'baʊt] I. prp. 1. um, um... herum;
2. umher in (dat.): **wander about the streets**;
3. bei, auf (dat.), an (dat.), um, in (dat.): (**somewhere**)
about the house irgendwo im Haus; **have you any**
money about you? haben Sie Geld bei sich?; **look**
about you! sieh dich um!; **there is nothing special**
about him an ihm ist nichts Besonderes;
4. wegen, über (acc.), um (acc.), von: **talk about**
business über Geschäfte sprechen; **I'll see about it**
ich werde danach sehen od. mich darum kümmern;
what is it about? worum handelt es sich?;
5. im Begriff, da bei: **he was about to go out**;
6. beschäftigt mit: **what is he about?** was macht er
(da)?; **he knows what he is about** er weiß, was er tut'

We distinguish between **internal** and **external** **software quality factors.**

Software Quality - Internal Quality Factors | 23

- Internal quality factors

- modular
- readable
- ...



This lecture series' main subject

An example of missing internal quality.

```
/// <summary>
/// Turns true into false and false into true
/// <param name="_booInpt">True or false</param>
/// <returns>False or true</returns>
private bool trueandorfalse(bool _booInpt)
{
    // I'm quite sure though there is a very
    // clever C# standard command doing this,
    // I just can't find it right now ...
    if (_booInpt == true)
        return false;
    return true;
}
```

An example of missing internal quality.

```
/**  
 * Checks to see if Australia is typed into the other country box  
 */  
function checkContactCountry(inputBox)  
{  
    var validator = new RegExp(  
        /^(A|a)(U|u)(S|s)(T|t)(R|r)(A|a)(L|l)(I|i)(A|a)  
        |(N|n)(E|e)(W|w)(Z|z)(E|e)(A|a)(L|l)(A|a)(N|n)(D|d)  
        |(N|n)(E|e)(W|w) (Z|z)(E|e)(A|a)(L|l)(A|a)(N|n)(D|d)$/);  
  
    if(validator.test(inputBox.value))  
    {  
        alert("Your Residential Address must be outside Australia."  
            + "Enter your residential address outside this country,"  
            + "or visit redacted-travel.com.au to make a booking if "  
            + "you live in Australia.");  
        inputBox.focus();  
        inputBox.select();  
    }  
}
```

Internal quality.

```
def isAnnotatedWith(  
    classFile: ClassFile,  
    annotationTypes: Iterable[ObjectType]): Boolean = {  
  
    var bufferOutput: Iterable[Object] = Iterable.empty  
    val runtimeVisibleAnnotations = classFile.runtimeVisibleAnnotations  
    val runtimeInvisibleAnnotations = classFile.runtimeInvisibleAnnotations  
    for (annotationType ← annotationTypes) {  
        bufferOutput = bufferOutput ++ runtimeVisibleAnnotations.filter {  
            case Annotation(`annotationType`, _) ⇒ true  
            case _ ⇒ false  
        }  
        bufferOutput = bufferOutput ++ runtimeInvisibleAnnotations.filter {  
            case Annotation(`annotationType`, _) ⇒ true  
            case _ ⇒ false  
        }  
    }  
  
    annotationTypes.nonEmpty &&  
    !classFile.isAnnotationDeclaration &&  
    bufferOutput.nonEmpty  
}
```

Where is the issue/are the issues?

“ [...] Have you ever noticed that **when someone checks in some complex and, oftentimes, horrific piece of code, the check-in is greeted with an almost deafening silence?** [...]”

The explanation for why this occurs was first given by C. Northcote Parkinson [...] He stated that if you were building something complex, then few people would argue with you because few people could understand what you were doing. If you were building something simple [...] which most anyone could build, then everyone would have an opinion.

Just one reason for “bad code”...

George V. Neville-Neil

Painting the Bike Shed - A sure-fire technique for ending pointless coding debates; ACM Queue, ACM 2009

1542-7730/09/0600

If you want to study code with missing quality...

Internal Quality Factors | 28

The screenshot shows a web browser window for thedailywtf.com. The header includes links for FEATURE ARTICLES, CODESOD, ERROR'D, FORUMS, and OTHER ARTICLES. A prominent advertisement for Puppet Labs features the text "New! Application Orchestration Eliminate IT complexity" and a link to puppetlabs.com. Below the ad, a sidebar on the left contains a question "WTF is The Daily WTF?" and a brief history: "Founded in 2004 by [Alex Papadimoulis](#), The Daily WTF is your how-not-to guide for developing software. We recount tales of disastrous development, from project". The main content area displays an article titled "CONFESION: RECT.CONTAINS(POINT)" by Remy Porter in CodeSOD on 2015-11-11.

But, reading other people's code - in particular if the code is good - is one of the best ways to learn to program.

The screenshot shows a continuation of the The Daily WTF website. On the left sidebar, there is a "Classic Articles" section with links to "The Brilliant Paula Bean", "Special Delivery", "Radio WTF: Make It Work", "ITAPPMONROBOT", and "Add your favorite...". There is also a link to "Article Archives". The main content area displays an article titled "SINGLE-USER MODE" by Erik Gern in Feature Articles on 2015-11-10. The article text discusses a brother named Andrzej who worked as a data-entry specialist for the Polish Ministry of Employment, while Jarosław was part of the ministry's IT department and moonlighted as a Python programmer. It quotes Jarosław saying, "No sane person can do this job," and Andrzej replying, "There just aren't enough hours in the day." At the bottom of the article, there are links for 111 Comments and Last Comment @ 07:16, along with social sharing icons for Facebook, Twitter, Email, and Google+.

If you want to study code...

The Scala programming language <http://www.scala-lang.org/>

24,723 commits	8 branches	102 releases	230 contributors
Branch: 2.11.x	scala / +		
SethTisue Merge pull request #4833 from xuwei-k/patch-2		Latest commit e10413e 2 days ago	
META-INF	Merge branch '2.10.x'	3 years ago	
doc	bump copyright year to 2015	4 months ago	
docs	Fix typos in spec, docs and comments	3 months ago	
lib	No longer support unreleased STARR.	2 years ago	
project	upgrade sbt from 0.13.7 to 0.13.9	2 months ago	
scripts	Windows CI: don't hardcode Ant path quite so hard	a month ago	
spec	"macro" is a reserved word since Scala 2.11	9 days ago	
src	Merge pull request #4803 from janekdb/2.11.x-conform-foreach-tparam	15 days ago	
test	SI-4950 Test reduction	17 days ago	
tools	Windows: make get-scala-commit-sha/date scripts work on Cygwin	a month ago	
.gitattributes	SI-9472 make Git use LF line endings on Windows	2 months ago	
.gitignore	Avoid wildcard ignorance of files named 'target'.	7 months ago	
.mailmap	update mailmap	2 years ago	
.travis.yml	opt-in to Travis's newer/faster container-based infrastructure	4 months ago	
CONTRIBUTING.md	Sync commit advice to README.md and fix Scabot URL	3 months ago	
Gemfile	use newer Redcarpet to build spec	4 months ago	

Mozilla Core 0 commits

billionairegenius 1019 commits

Software quality in commercial software.

Part of the source code for Comanche, build 055.
It is part of the source code for the Command
Module's (CM) Apollo Guidance Computer (AGC),
Apollo 11.

SET EB

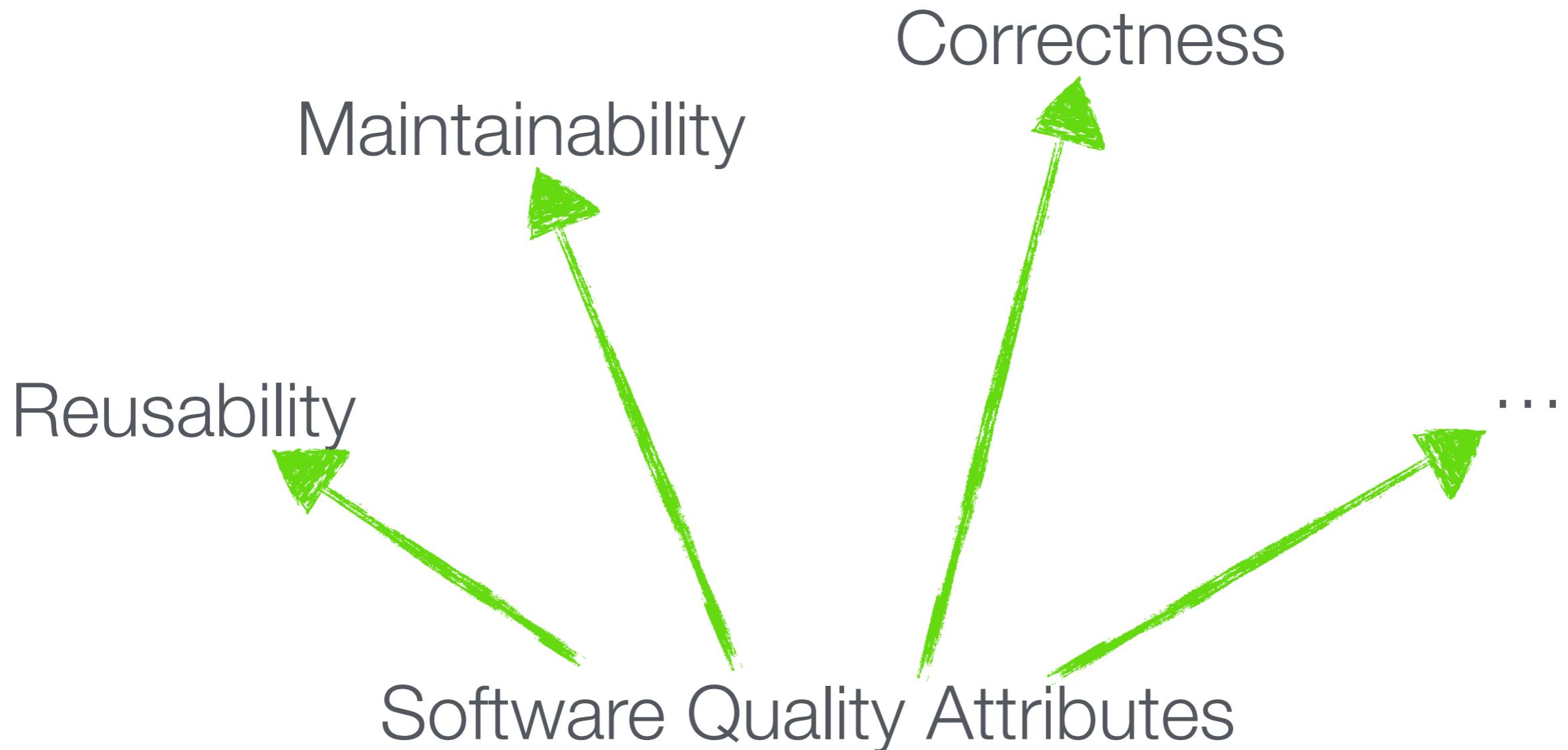
EGEXIT.

/ .3048) /2VS

```
50      VXSC    PDVL
51          -KVSCALE      # KVSCALE = .81491944
52          UNITW       # FULL UNIT VECTOR
53      VXV     VXSC      # VREL = V - WE*R
54          UNITR
55          KWE
56      VAD     STADR
57      STORE   -VREL      # SAVE FOR ENTRY GUIDANCE.      REF COORDS
58
59      UNIT    LXA,1
60          36D       # ABVAL( -VREL) TO X1
61      STORE   UXA/2      # -UVREL      REF COORDS
62
63      VXV     VCOMP
64          UNITR      # .5 UNIT      REF COORDS
65      UNIT    SSP       # THE FOLLOWING IS TO PROVIDE A STABLE
66          S1        # UN FOR THE END OF THE TERMINAL PHASE.
67  SPVQUIT DEC     .019405 # 1000/ 2 VS
68          TIX,1    VLOAD      # IF V-VQUIT POS, BRANCH.
69          CM/POSE2 # SAME UYA IN OLDDUYA
```

It is often not possible to improve all software quality attributes.

Sometimes they are at odds.



Software Quality

- **Good Software**

Ian Sommerville; Software Engineering - Eighth
Edition; Addison Wesley, 2007



TECHNISCHE
UNIVERSITÄT
DARMSTADT

- **Maintainability**

Software should be written in such a way that it may evolve to meet changing needs of customers.

- **Efficiency**

Software should not waste system resources; it includes: responsiveness, processing time, memory utilisation, etc.

- **Usability**

Software must be usable by the intended users.

- **Dependability (dt. Verlässlichkeit)**

Dependable software does not cause physical or economic damage in the event of system failure. Further properties: Repairability, Survivability, Error Tolerance...

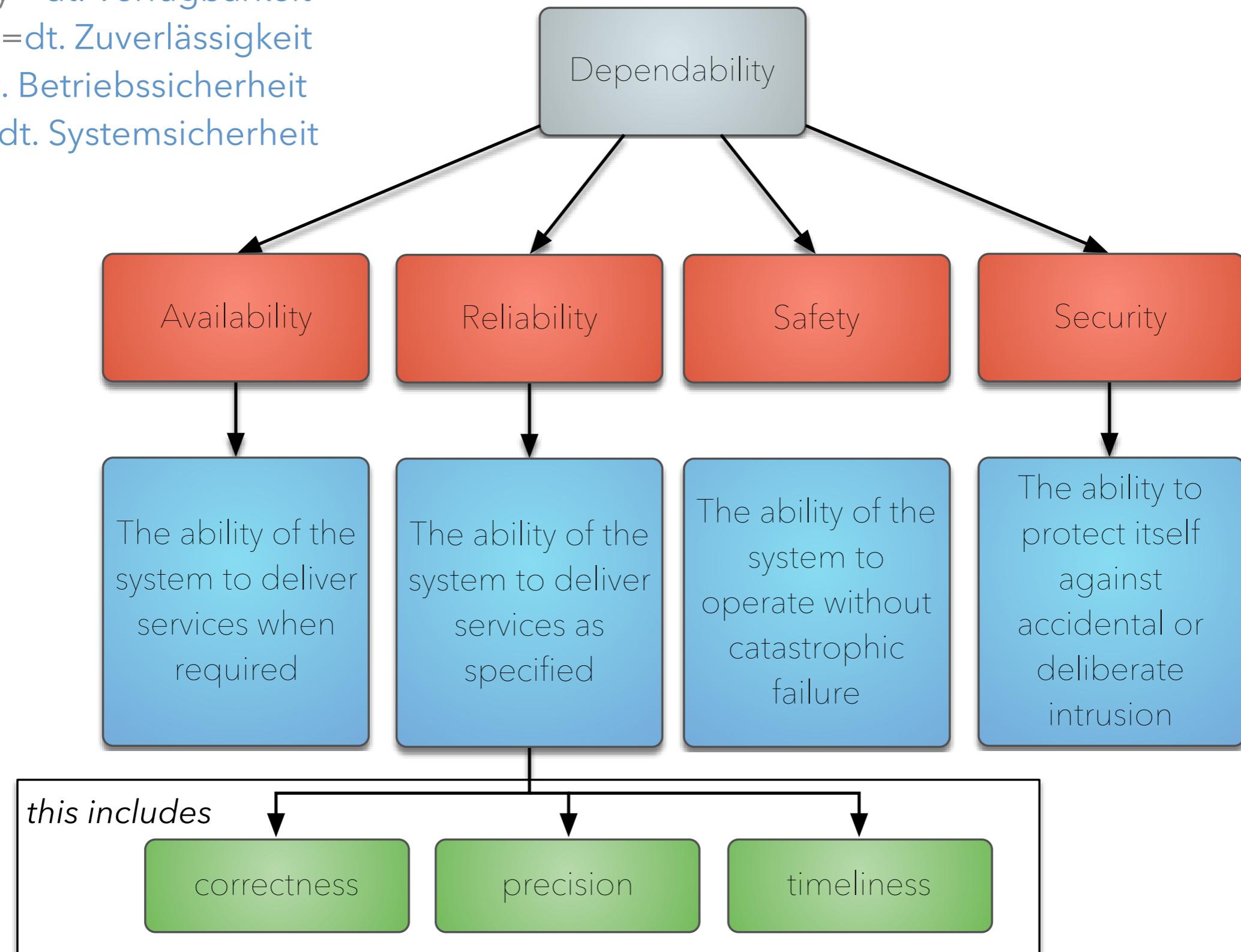
Some Aspects of Dependable Systems

Availability = dt. Verfügbarkeit

Reliability = dt. Zuverlässigkeit

Safety = dt. Betriebssicherheit

Security = dt. Systemsicherheit



Software Quality Assurance (SQA)

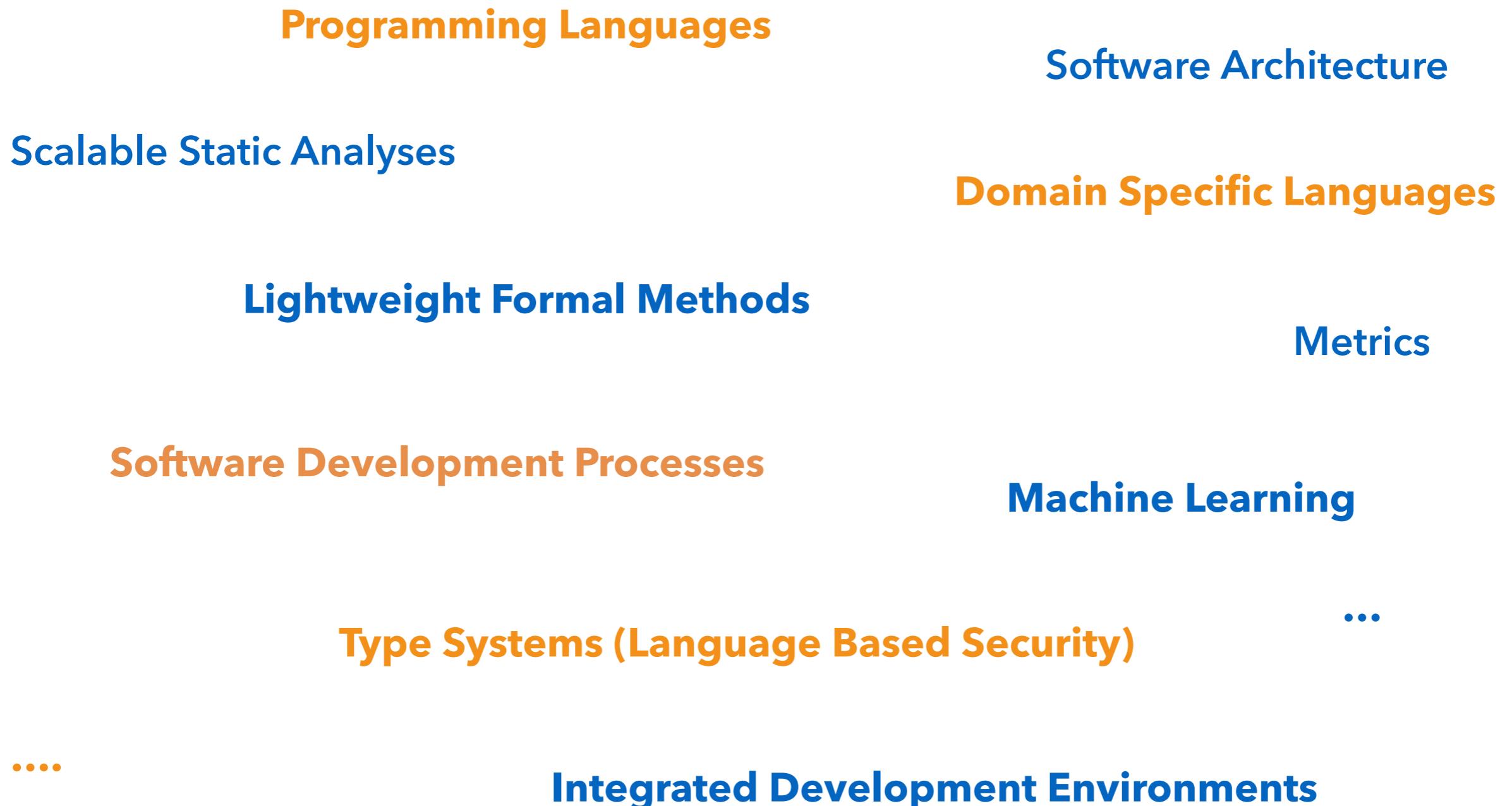
- Constructive vs. Analytical



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Fostering Software Quality by Means of... Constructive SQA and Analytical SQA

Software Quality Assurance | 36



Null Values...

- found in *java.nio.file.FileTreeWalker next()*

```
if (ioe != null) {  
    ioe = e;  
} else {  
    ioe.addSuppressed(e);  
}
```

here, ioe is null



A Few (Well Known) Static Analysis Tools

Fostering Software Quality | 39

- FindBugs/SpotBugs
Lightweight static analyses on top of Java Bytecode.
- PMD
Lightweight static analyses on top of the AST using Java Visitors or XPath based rules.
- CheckStyle
Lightweight static analyses on top of the AST using Java Visitors.
- CheckerFramework
Static analyses using pluggable types.
- ConQAT
Code Clone Detection.



FindBugs™ – Find Bugs in Java Programs

http://findbugs.sourceforge.net/index.html

Google

UNIVERSITY OF MARYLAND 1856

TM

UNIVERSITY OF MARYLAND 1856

FindBugs™ - Find Bugs in Java Programs

This is the web page for FindBugs, a program which uses static analysis to look for bugs in Java code. It is free software, distributed under the terms of the [Lesser GNU Public License](#). The name FindBugs™ and the [FindBugs logo](#) are trademarked by [The University of Maryland](#). As of July, 2008, FindBugs has been downloaded more than 700,000 times.

FindBugs requires JRE (or JDK) 1.5.0 or later to run. However, it can analyze programs compiled for any version of Java. The current version of FindBugs is 1.3.9, released on 20:11:47 EDT, 21 August, 2009. [We are very interested in getting feedback on how to improve FindBugs.](#)

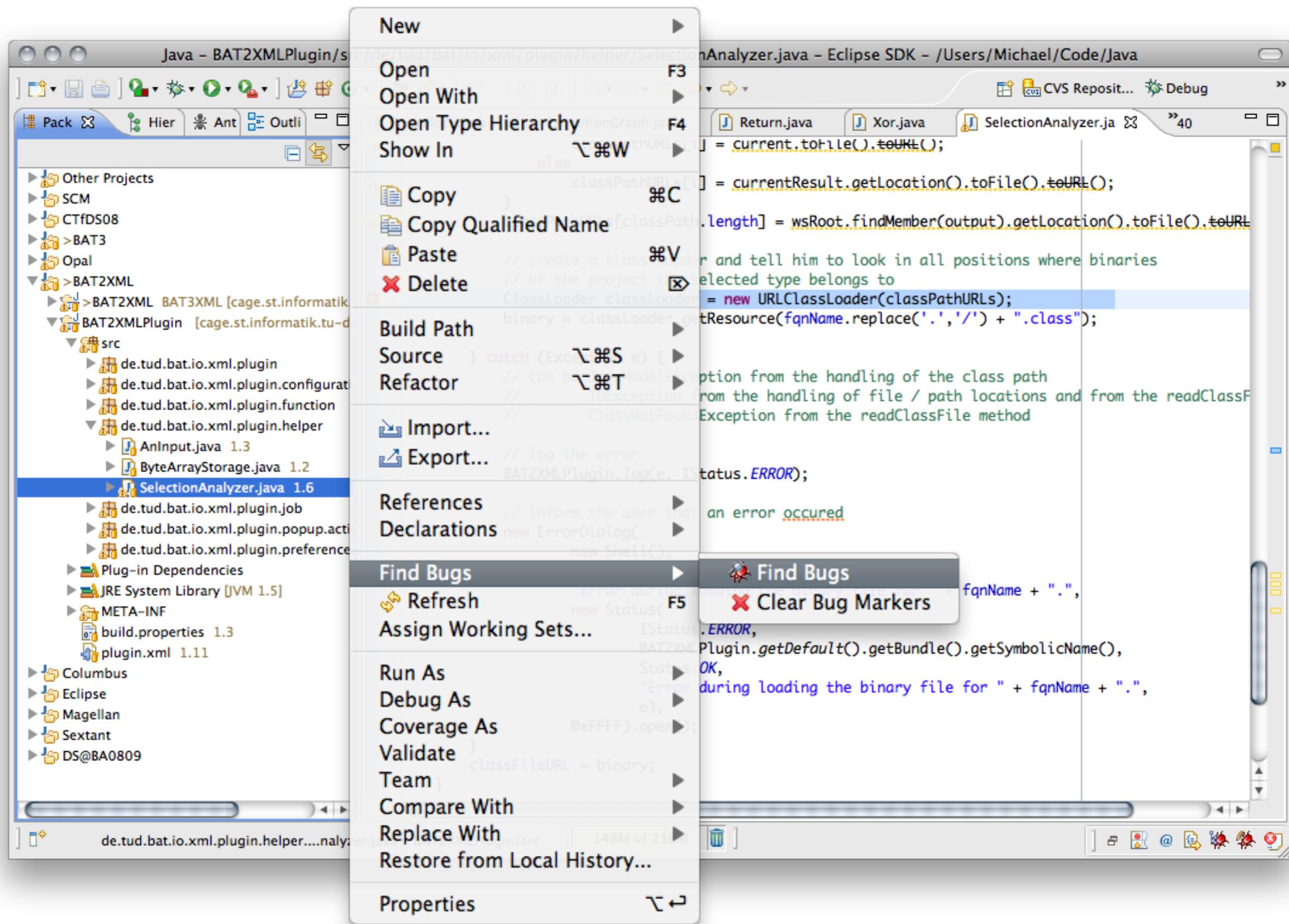
[Changes](#) | [Talks](#) | [Papers](#) | [Sponsors](#) | [Support](#)

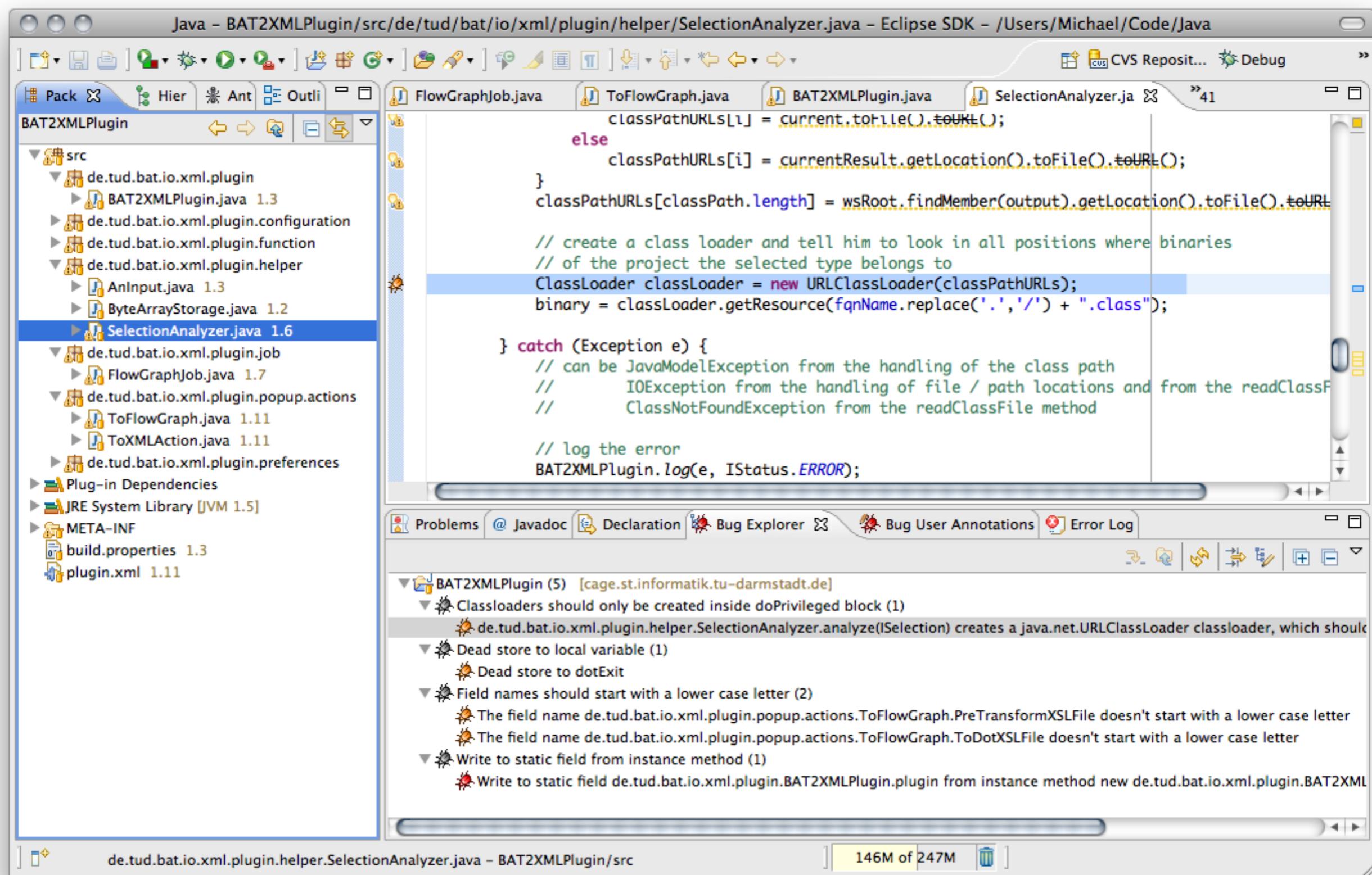
New

- **JavaOne talk:** [Slides](#) from my JavaOne talk, Mistakes That Matter.
- **FindBugs community review:** We are previewing FindBugs community review, in which anyone can review issues in open source projects (i.e., mark issues as "must fix" or "mostly harmless"), and those reviews are automatically shared with other reviewers.

This is a pre-beta release, not ready for deployment. The implementation will be undergoing significant changes before general availability.

Initially, we are posting results for:







Fork me on GitHub

PMD is a source code analyzer. It finds common programming flaws like unused variables, empty catch blocks, unnecessary object creation, and so forth. It supports Java, JavaScript, XML, XSL.

Additionally it includes CPD, the copy-paste-detector. CPD finds duplicated code in Java, C, C++, C#, PHP, Ruby, Fortran, JavaScript.

Latest version

Get Involved

Plugins

Recent Announcements

Next development version

Previous versions

Latest version

5.2.1 (3rd November 2014)

- [Release Notes](#)
- [Download \(Sourcecode, Documentation\)](#)
- [Online Documentation](#)

Checkstyle 6.1



Last Published: 2014-11-13 | Version: 6.1

About

[Checkstyle](#)
[Release Notes](#)

Documentation

- ▼ [Configuration](#)
Property Types
- ▼ [Running](#)
Ant Task
Command Line
- [Available Checks](#)
- ▼ [Standard Checks](#)
Annotations
Block Checks
Class Design
Coding
Duplicate Code
Headers
Imports
Javadoc Comments
Metrics
Miscellaneous
Modifiers
Naming Conventions
Regexp
Size Violations
Whitespace
- ▼ [Extending Checkstyle](#)
Writing checks
Writing filters
Writing listeners
- ▼ [Style Configurations](#)
Google's Style
Sun's Style

Developers

[Javadoc](#)
[Project Page](#) ↗
[Contributing](#)

Project Documentation

- ▼ [Project Information](#)

Overview

Checkstyle is a development tool to help programmers write Java code that adheres to a coding standard. It automates the process of checking Java code to spare humans of this boring (but important) task. This makes it ideal for projects that want to enforce a coding standard.

Checkstyle is highly configurable and can be made to support almost any coding standard. An example configuration files are supplied supporting the [Sun Code Conventions](#) ↗, [Google Java Style](#) ↗.

A good example of a report that can be produced using Checkstyle and [Maven](#) ↗ can be seen [here](#) ↗.

Important Development Changes

As of September 2013, the Checkstyle project is using GitHub for hosting the following:

- [Source code repository](#) ↗ - replacing the Mercurial repository on SourceForge.
- [Issue management](#) ↗ - replacing the Bugs/Feature/Patches on SourceForge. All new issues should be raised at GitHub, and pull requests are now the preferred way to submit patches.

SourceForge will still be used for website hosting and binary hosting for downloads.

Project Analysis Help

Number of issues currently displayed: [Relevance ≥ 75] 837(Total issues: 7876) Search:

▼ JDK 1.8.0_66

- ▶ Project Files
 - ▶ apple.applescript
 - ▶ apple.laf
 - ▶ apple.launcher
 - ▶ apple.security
 - ▶ com.apple.concurrent
 - ▶ com.apple.eawt
 - ▶ com.apple.eawt.event
 - ▶ com.apple.eio
 - ▶ com.apple.laf
 - ▶ com.apple.laf.resources
 - ▶ com.oracle.deploy.update
 - ▶ com.oracle.jrockit.jfr
 - ▶ com.oracle.jrockit.jfr.client
 - ▶ com.oracle.jrockit.jfr.manager
 - ▶ com.oracle.net
 - ▶ com.oracle.nio
 - ▶ com.oracle.util
 - ▶ com.oracle.webservices.interr
 - ▶ com.oracle.xmlns.internal.web
 - ▶ com.sun.accessibility.internal
 - ▶ com.sun.activation.registries
 - ▶ com.sun.applet2
 - ▶ com.sun.applet2.preloader
 - ▶ com.sun.applet2.preloader.ev
 - ▶ com.sun.awt
 - ▶ com.sun.beans
 - ▶ com.sun.beans.decoder
 - ▶ com.sun.beans.editors
 - ▶ com.sun.beans.finder
 - ▶ com.sun.beans.infos
 - ▶ com.sun.beans.util

▼ javax.security.auth.kerberos (Issues: 1)

class [public final \[SUPER\] ServicePermission](#)
 method [private static String getActions \(int\)](#)
 instruction [pc=18 line=248](#)

[dead code] the successor instruction is dead: pc=21 (line=248)
 relevance 99 (of utmost relevance)

summary [▲](#) The evaluation of the instruction never leads to the evaluation of the specified instruction.

if int = 0 == 0

▼ javax.sound.sampled (Issues: 2)

class [public \[SUPER\] AudioFileFormat\\$Type](#)
 method [public final boolean equals \(Object\)](#)
 instruction [pc=8 line=381](#)

guard
[pc=36 line=384](#)
 unguarded access

relevance 99 (of utmost relevance)

summary [▲](#) Unguarded local variable access ({_ <: java.lang.Object, null}[@-2;t=102]) though explicit test is done elsewhere.

class [public \[SUPER\] AudioFormat\\$Encoding](#)
 method [public final boolean equals \(Object\)](#)
 instruction [pc=8 line=639](#)

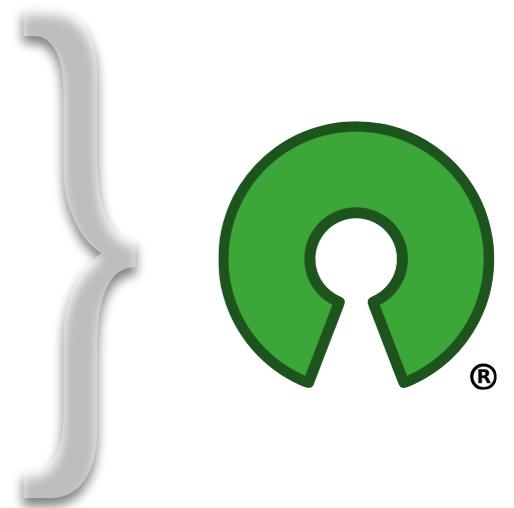
guard

BugPicker Log	Project Log	Source code	Bytecode
314		iflt	328
317	1862	aload_0	
318		getfield	..FilePane\$Handler { ..FilePane this\$0 }
321		iload_3	
322		invokestatic	..FilePane { void access\$3200 (..FilePane, int) }
325		goto	353
328	1864	aload_0	
329		getfield	..FilePane\$Handler { ..FilePane this\$0 }
332		invokestatic	..FilePane { void access\$3300 (..FilePane) }
335		goto	353
338	1867	aload_1	
339		invokevirtual	java.awt.event.MouseEvent { int getClickCount () }
		incost ?	

A Few (Well Known) Static Analysis Tools

Fostering Software Quality | 46

- JDepend
Structural analysis on top of Java Bytecode.
- DependencyFinder
Structural analysis on top of Java Bytecode.
- Stan4J
Structural analysis on top of Java Bytecode.
- Sonargraph (SonarJ)
Analyzes the structure of applications.



A Few (Well Known) Static Analysis Tools

Fostering Software Quality | 47

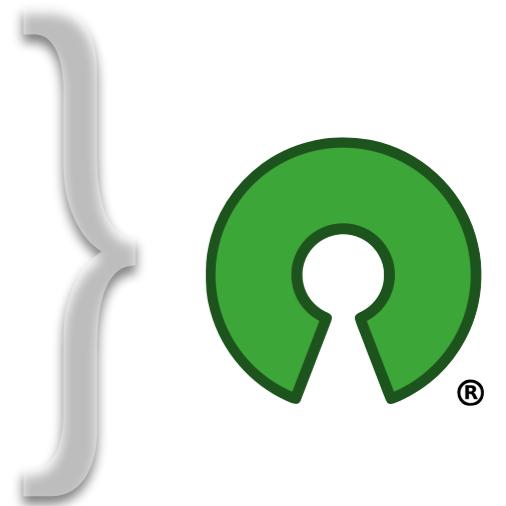
- ESC/Java2

Formal verification using JML Annotations.

- Key

Formal verification.

- ...



	reusability	maintainability	correctness	effort
lightweight static analyses	code clone detection	✓	✓	↓ -○
semi formal methods			✓	↓
formal methods			✓	↑
structure analyses	✓	✓		↓
style conformance checking		✓		↓
architecture conformance checking	✓	✓		○ - ↑

violations of best practices/bug patterns

Classifying Found Issues

Software Quality - Terminology | 49

- **True** and **False** Positives
- **True** and **False** Negatives
- **Irrelevant True** Positives
- **Perceived False** Positives

Classification of issues identified by static analysis tools.

True and False Positives

Software Quality - Terminology | 50

- a **True** Positive is the correct finding (*of something relevant*)

This is what static analyses should detect.

- a **False** Positive is a finding that is just incorrect

False positives are typically caused by the weaknesses of the analysis.

Classification of issues identified by static analysis tools.

Example of a **True** Positive

Software Quality - Terminology | 51

Let's assume that we have a "basic" analysis to detect object accesses (`o.xyz`) that appear in a guarded context (`if (o != null)`) and also outside a guarded context.

The diagram shows a Java code snippet with annotations. A green curved arrow labeled "guard" points from the `if (args != null)` guard to the `System.out.println("number of elements: " + args.length);` statement. A red curved arrow labeled "guarded & unguarded access" points from the `System.out.println(arg);` statement back to the `if (args != null)` guard. The code is as follows:

```
void printIt(String args[]) {  
    if (args != null) {  
        System.out.println("number of elements: " + args.length);  
    }  
    for (String arg : args) {  
        System.out.println(arg);  
    }  
}
```

Classification of issues identified by static analysis tools.

Example of a **False** Positive

Software Quality - Terminology | 52

Let's assume that we have a "basic" analysis to detect object accesses (`o.xyz`) that appear in a guarded context (`if (o != null)`) and also outside a guarded context.

The diagram shows the `printReverse` method with several annotations:

- A green circle labeled "guard" surrounds the `if (args != null)` guard clause.
- A red circle labeled "guarded & unguarded access" surrounds the `args.length` access, which is highlighted with a red arrow from the "guard" annotation.
- An orange circle labeled "implicit guard" surrounds the `System.out.println(args[i])` call, which is highlighted with an orange arrow from the "guard" annotation.

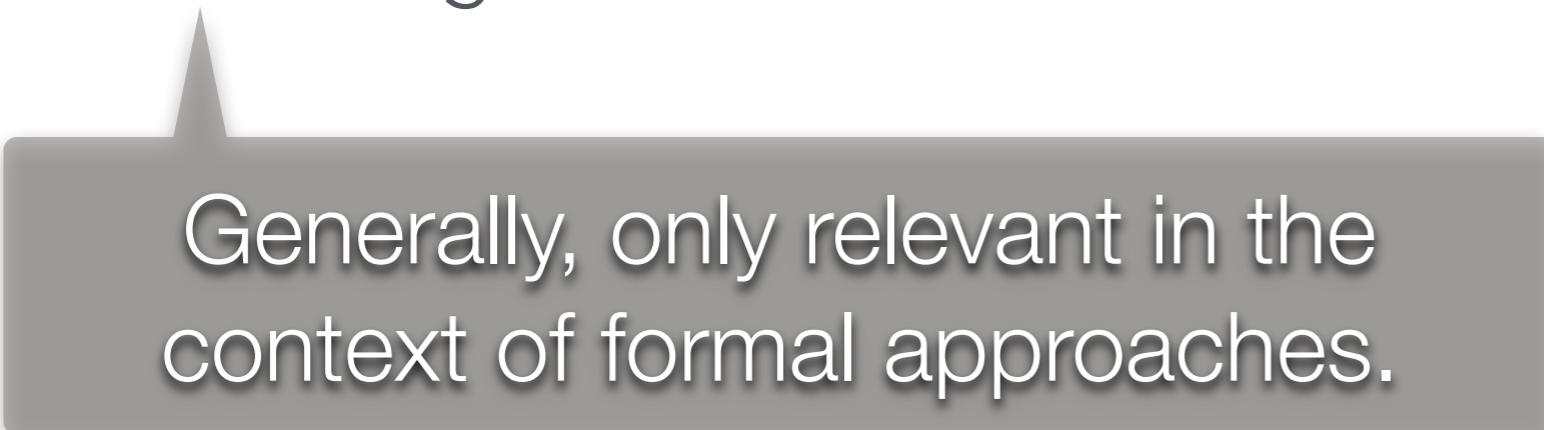
```
void printReverse(String args[]) {
    int argcnt = 0;
    if (args != null) {
        argcnt = args.length;
    }
    for (int i = argcnt - 1, i >= 0; argcnt--) {
        System.out.println(args[i]);
    }
}
```

Classification of issues identified by static analysis tools.

True and **False** Negatives

Software Quality - Terminology | 53

- a **True** Negative is the correct finding of no issue.
- a **False** Negative is an issue that is not reported.



Generally, only relevant in the context of formal approaches.

Classification of issues identified by static analysis tools.

Irrelevant True Positives

Software Quality - Terminology | 54

- Irrelevancy is context-dependent...
 - Issues related to Serialization are irrelevant when your application doesn't use Serialization at all.
 - A violation of the `hashCode-equals` contract may be completely irrelevant for an (inner) class that is never put in a collection that uses hashes.
 - ...



Classification of issues identified by static analysis tools.

Irrelevant True Positives

Software Quality - Terminology | 55

```
boolean handleIt(int i) {  
    if (i < 0 || i > 2)  
        throw new IllegalArgumentException();  
  
    switch (i) {  
        case 0:  
        case 1:  
            return true;  
  
        case 2:  
            return false;  
  
        default:  
            throw new UnknownError();  
    }  
}
```

- They are typically related to:
- default cases in switch statements
 - assertions
 - a test that leads to an AssertionError



Classification of issues identified by static analysis tools.

Perceived False Positives

Software Quality - Terminology | 56

```
GeneralPath result = new GeneralPath(GeneralPath.WIND_NON_ZERO);  
...  
if (dx != 0 || dy != 0) {  
    AffineTransform tx = AffineTransform.getTranslateInstance(dx, dy);  
    result = (GeneralPath)tx.createTransformedShape(result);  
}
```

This cast will
always fail!

java.awt.font.TextLayout - Line 2404ff

Perceived False Positives

```
GeneralPath result = new GeneralPath(GeneralPath.WIND_NON_ZERO);
...
if (dx != 0 || dy != 0) {
    AffineTransform tx = AffineTransform.getTranslateInstance(dx, dy);
    result = (GeneralPath)tx.createTransformedShape(result);
}

public Shape createTransformedShape(Shape pSrc) {
    if (pSrc == null) {
        return null;
    }
    return new Path2D.Double(pSrc, this);
}
```



interface Shape

```
class Path2D implements Shape, Cloneable
java.awt.font.TextLayout - Line 2404ff
/*inner*/ class Double extends Path2D implements Serializable
```

Perceived False Positives

Software Quality - Terminology | 58

```
GeneralPath result = new GeneralPath(GeneralPath.WIND_NON_ZERO);
...
if (dx != 0 || dy != 0) {
    AffineTransform
    result = (
        public
        int
    }
    return new Path2D.Double(pSrc, this);
}
```

Perceived false positives are the result of issue reports related to complex issues and/or related to reports that are not easy to comprehend.

interface Shape

java.awt.font.TextLayout - Line 2404ff

/*inner*/ class Double extends Path2D implements Serializable

Classification of issues identified by static analysis tools.

Perceived False Positives

Software Quality - Terminology | 59

```
protected Icon getIconForType(int messageType) {  
    if(messageType < 0 || messageType > 3) return null;  
    String propertyName = null;  
    switch(messageType) {  
        case 0:  
            propertyName = "OptionPane.errorIcon"; break;  
        case 1:  
            propertyName = "OptionPane.informationIcon"; break;  
        case 2:  
            propertyName = "OptionPane.warningIcon"; break;  
        case 3:  
            propertyName = "OptionPane.questionIcon"; break;  
    }  
    if (propertyName != null) {  
        return (Icon)DefaultLookup.get(optionPane, this, propertyName);  
    }  
    return null;  
}
```



Dead Code!

javax.swing.plaf.basic.BasicOptionPaneUI

Classification of issues identified by static analysis tools.

Perceived False Positives

Software Quality - Terminology | 60

```
protected Icon getIconForType(int messageType) {  
    String propertyName = null;  
    switch(messageType) {  
        case 0:  
            propertyName = "OptionPane.errorIcon"; break;  
        case 1:  
            propertyName = "OptionPane.informationIcon"; break;  
        case 2:  
            propertyName = "OptionPane.warningIcon"; break;  
        case 3:  
            propertyName = "OptionPane.questionIcon"; break;  
        default:  
            return null;  
    }  
    return (Icon)DefaultLookup.get(optionPane, this, propertyName);  
}
```

javax.swing.plaf.basic.BasicOptionPaneUI

Classification of issues identified by static analysis tools.

Cryptic True Positives

Software Quality - Terminology | 61

```
boolean process() throws Exception {  
    boolean done = false;  
    do {  
        Thread.sleep(500);  
        done = (System.currentTimeMillis() % 100l == 0l);  
    } while (!done);  
    return !done;  
}
```

Dead Edge!

Refactored: return false;

```
if (done)  
    done = false  
else  
    done = true  
return done
```



A Holistic View is required.



Software Quality

- Summary



TECHNISCHE
UNIVERSITÄT
DARMSTADT

International Conference on Software Engineering

Proceedings of the 2009 IEEE 31st International Conference on Software Engineering

Year of Publication: 2009

ISBN:978-1-4244-3453-4

Does Distributed Development Affect Software Quality? An Empirical Case Study of Windows Vista

Christian Bird¹, Nachiappan Nagappan², Premkumar Devanbu¹, Harald Gall³, Brendan Murphy²

¹University of California, Davis, USA

²Microsoft Research

³University of Zurich, Switzerland

{cabird, ptdevanbu}@ucdavis.edu {nachin, bmurphy}@microsoft.com gall@ifi.uzh.ch

ACM Queue

Volume 7 , Issue 5 (June 2009)

Distributed Computing

Year of Publication: 2009

ISSN:1542-7730



Browser Security: Lessons from Google Chrome

Google Chrome developers focused on three key problems to shield the browser from attacks.

Charles Reis, Google; Adam Barth, UC Berkeley ; Carlos Pizano, Google

The Web has become one of the primary ways people interact with their computers, connecting people with a diverse landscape of content, services, and applications. Users can find new and interesting content on the Web easily, but this presents a security challenge: malicious Web-site operators can attack users through their Web browsers. Browsers face the challenge of keeping their users safe while providing a rich platform for Web applications.

Browsers are an appealing target for attackers because they have a large and complex trusted computing base with a wide network-visible interface. Historically, every browser at some point has contained a bug that let a malicious Web-site operator circumvent the browser's security policy and compromise the user's computer. Even after these vulnerabilities are patched, many users continue to run older, vulnerable versions.⁵ When these users visit malicious Web sites, they run the risk of having their computers compromised.

Generally speaking, the danger posed to users comes from three factors, and browser vendors can help keep their users safe by addressing each of these factors:

- **The severity of vulnerabilities.** By sandboxing their rendering engine, browsers can reduce the severity of vulnerabilities. Sandboxes limit the damage that can be caused by an attacker who exploits a vulnerability in the rendering engine.
- **The window of vulnerability.** Browsers can reduce this window by improving the user experience

Recall the “Fifteen Principles of Software Engineering”.

Take responsibility!

There are no excuses. If you develop a system, it is your responsibility to do it right. Take that responsibility. Do it right, or don’t do it at all.

The goal of this lecture is to enable you to systematically carry out small(er) software projects that produce quality software.

-
- Software quality is not just about the (internal) quality of the source code.
 - Software quality means different things to different stake holders.
 - To produce quality software a holistic view on a software project is required.

- The goal of this lecture is to enable you to systematically carry out small(er) commercial or open-source projects.

