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

Introduction to Software Engineering

Software Quality

TECHNISCHE
UNIVERSITÄT
DARMSTADT

Software Quality

- **Software Quality Factors**

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

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

- 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**.

- 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 of 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();
   }
}
```

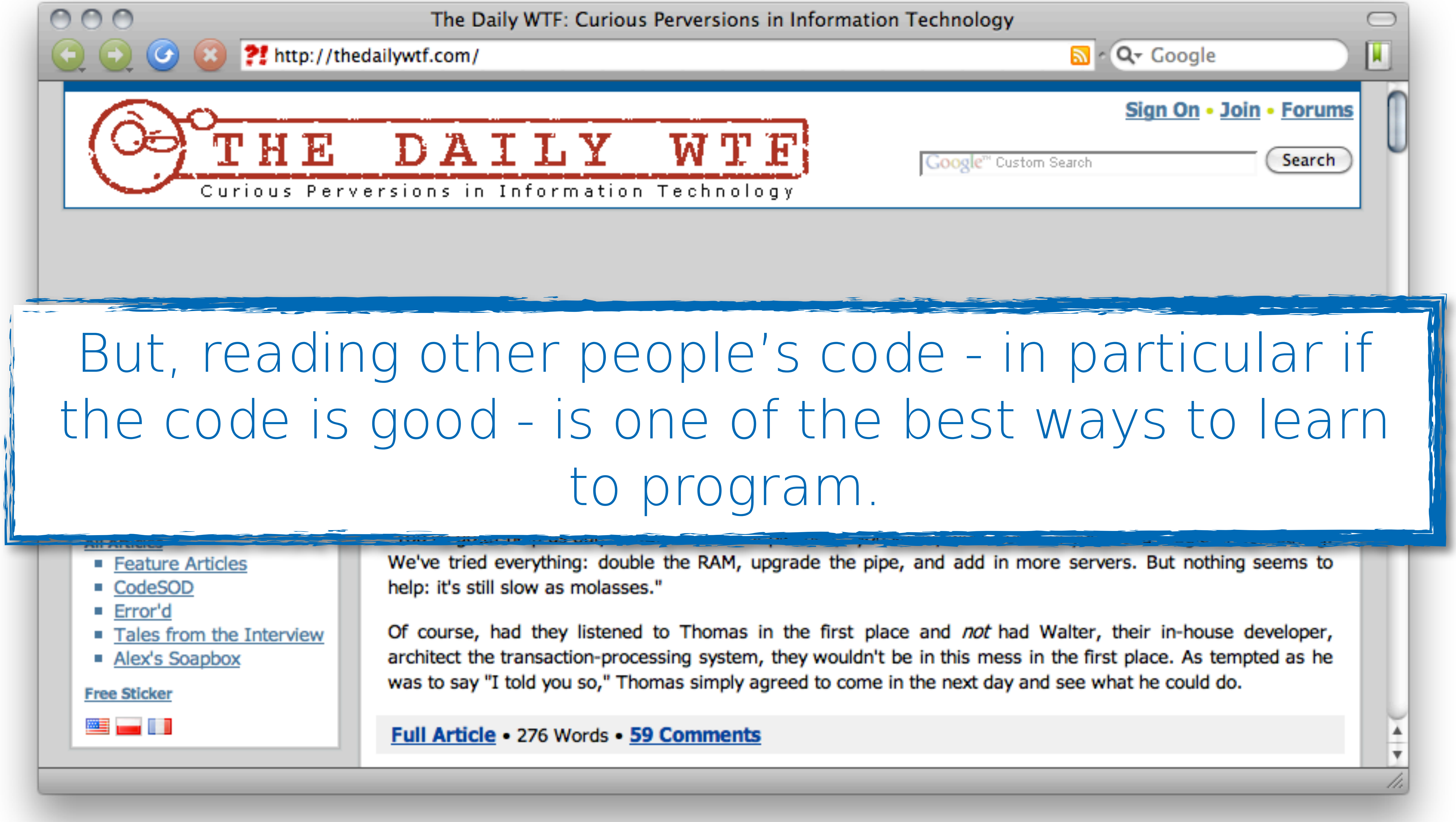
> *[…] Have you ever noticed that* ***when someone checks in some complex and, oftentimes, horrific piece of code, the check-in is greeted with an almost deafening silence?*** *[…]*
>
> *The explanation for why this occurs was first given by C. Northcote Parkinson […] He stated that if you were building 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...



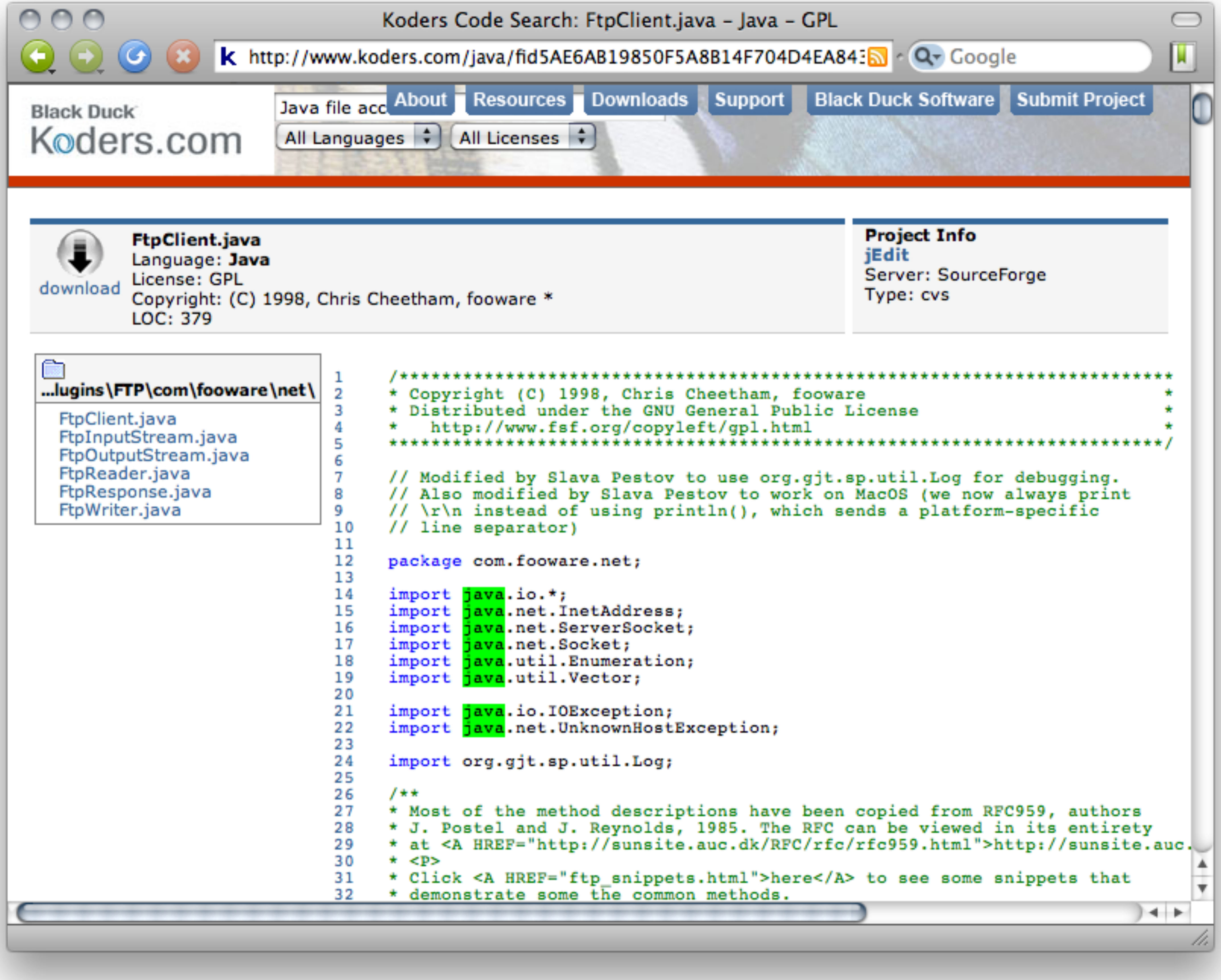
We've tried everything: double the RAM, upgrade the pipe, and add in more servers. But nothing seems to help: it's still slow as molasses."

Of course, had they listened to Thomas in the first place and *not* had Walter, their in-house developer, architect the transaction-processing system, they wouldn't be in this mess in the first place. As tempted as he was to say "I told you so," Thomas simply agreed to come in the next day and see what he could do.

Full Article • 276 Words • **59 Comments**

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

If you want to study code...



FtpClient.java
Language: **Java**
License: GPL
Copyright: (C) 1998, Chris Cheetham, fooware *
LOC: 379

Project Info
jEdit
Server: SourceForge
Type: cvs

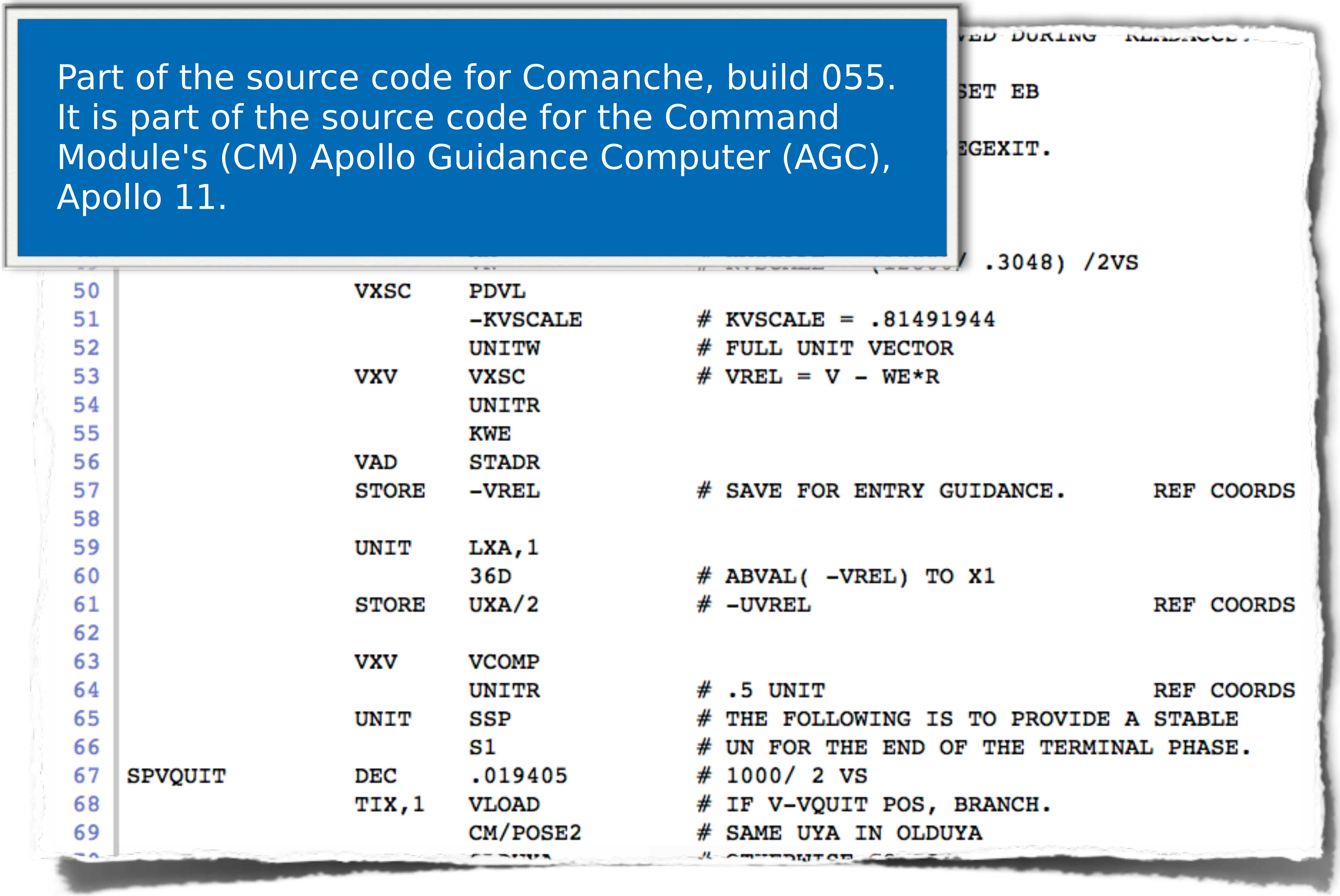
...lugins\FTP\com\fooware\net\

- FtpClient.java
- FtpInputStream.java
- FtpOutputStream.java
- FtpReader.java
- FtpResponse.java
- FtpWriter.java

```
/*************************************************************************
 * Copyright (C) 1998, Chris Cheetham, fooware                          *
 * Distributed under the GNU General Public License                     *
 *   http://www.fsf.org/copyleft/gpl.html                               *
 *************************************************************************/

// Modified by Slava Pestov to use org.gjt.sp.util.Log for debugging.
// Also modified by Slava Pestov to work on MacOS (we now always print
// \r\n instead of using println(), which sends a platform-specific
// line separator)

package com.fooware.net;

import java.io.*;
import java.net.InetAddress;
import java.net.ServerSocket;
import java.net.Socket;
import java.util.Enumeration;
import java.util.Vector;

import java.io.IOException;
import java.net.UnknownHostException;

import org.gjt.sp.util.Log;

/**
 * Most of the method descriptions have been copied from RFC959, authors
 * J. Postel and J. Reynolds, 1985. The RFC can be viewed in its entirety
 * at <A HREF="http://sunsite.auc.dk/RFC/rfc/rfc959.html">http://sunsite.auc.
 * <P>
 * Click <A HREF="ftp_snippets.html">here</A> to see some snippets that
 * demonstrate some the common methods.
```

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.



```
                                             ED DURING  READACCEL.

                                             SET EB

                                             EGEXIT.




                                             .3048) /2VS
               VXSC     PDVL
                        -KVSCALE        # KVSCALE = .81491944
                        UNITW           # FULL UNIT VECTOR
               VXV      VXSC            # VREL = V - WE*R
                        UNITR
                        KWE
               VAD      STADR
               STORE    -VREL           # SAVE FOR ENTRY GUIDANCE.      REF COORDS

               UNIT     LXA,1
                        36D             # ABVAL( -VREL) TO X1
               STORE    UXA/2           # -UVREL                        REF COORDS

               VXV      VCOMP
                        UNITR           # .5 UNIT                       REF COORDS
               UNIT     SSP             # THE FOLLOWING IS TO PROVIDE A STABLE
                        S1              # UN FOR THE END OF THE TERMINAL PHASE.
SPVQUIT        DEC      .019405         # 1000/ 2 VS
               TIX,1    VLOAD           # IF V-VQUIT POS, BRANCH.
                        CM/POSE2        # SAME UYA IN OLDUYA
```

http://googlecode.blogspot.com/2009/07/apollo-11-missions-40th-anniversary-one.html

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

- 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.

- 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.

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

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

- 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

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

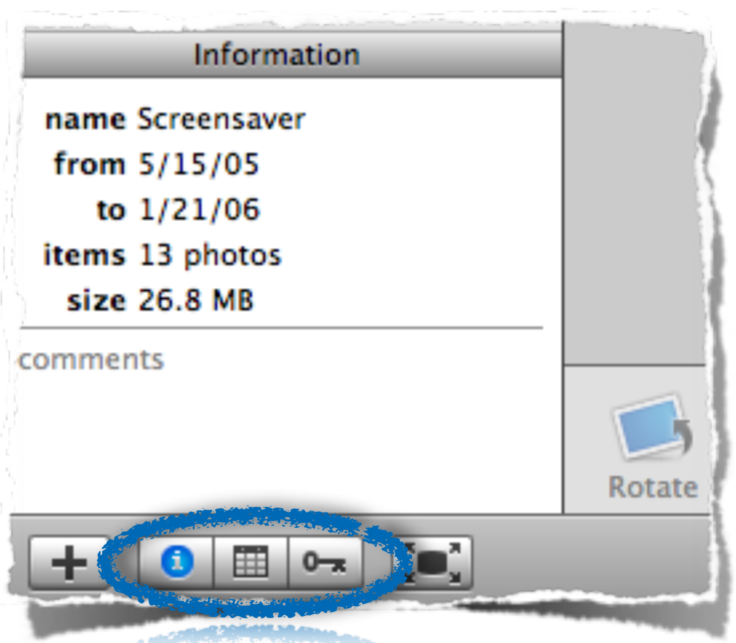
- Resources are the processor time, the space occupied in internal and external memories, the bandwidth used in communication devices, ……
- Always try to use "good" algorithms over "bad" ones, because a computer that is twice as fast as a previous model can handle problem sizes near 2*N if the algorithm's 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.

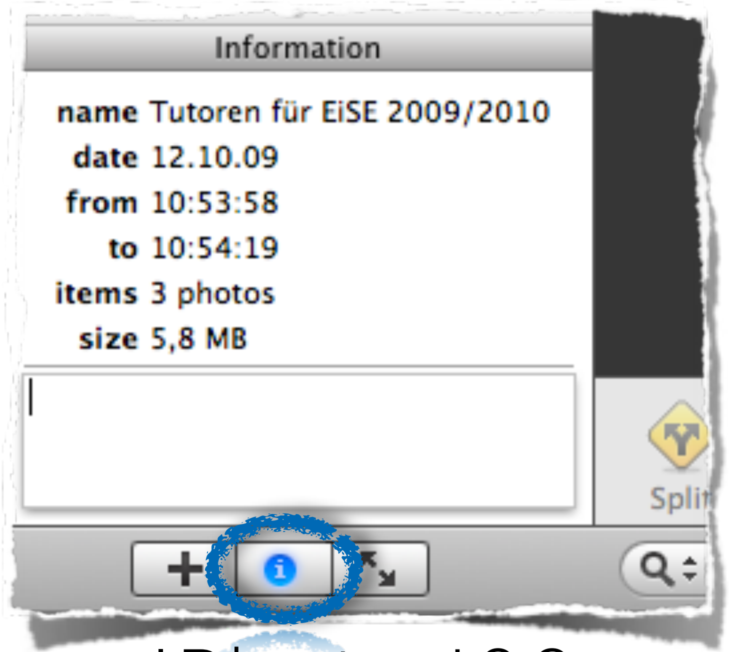
- Avoid featurism; remain consistent with existing features if you add new ones

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.



name Screensaver
from 5/15/05
to 1/21/06
items 13 photos
size 26.8 MB

IPhoto ‘06



name Tutoren für EiSE 2009/2010
date 12.10.09
from 10:53:58
to 10:54:19
items 3 photos
size 5,8 MB

IPhoto ’09

Software Quality

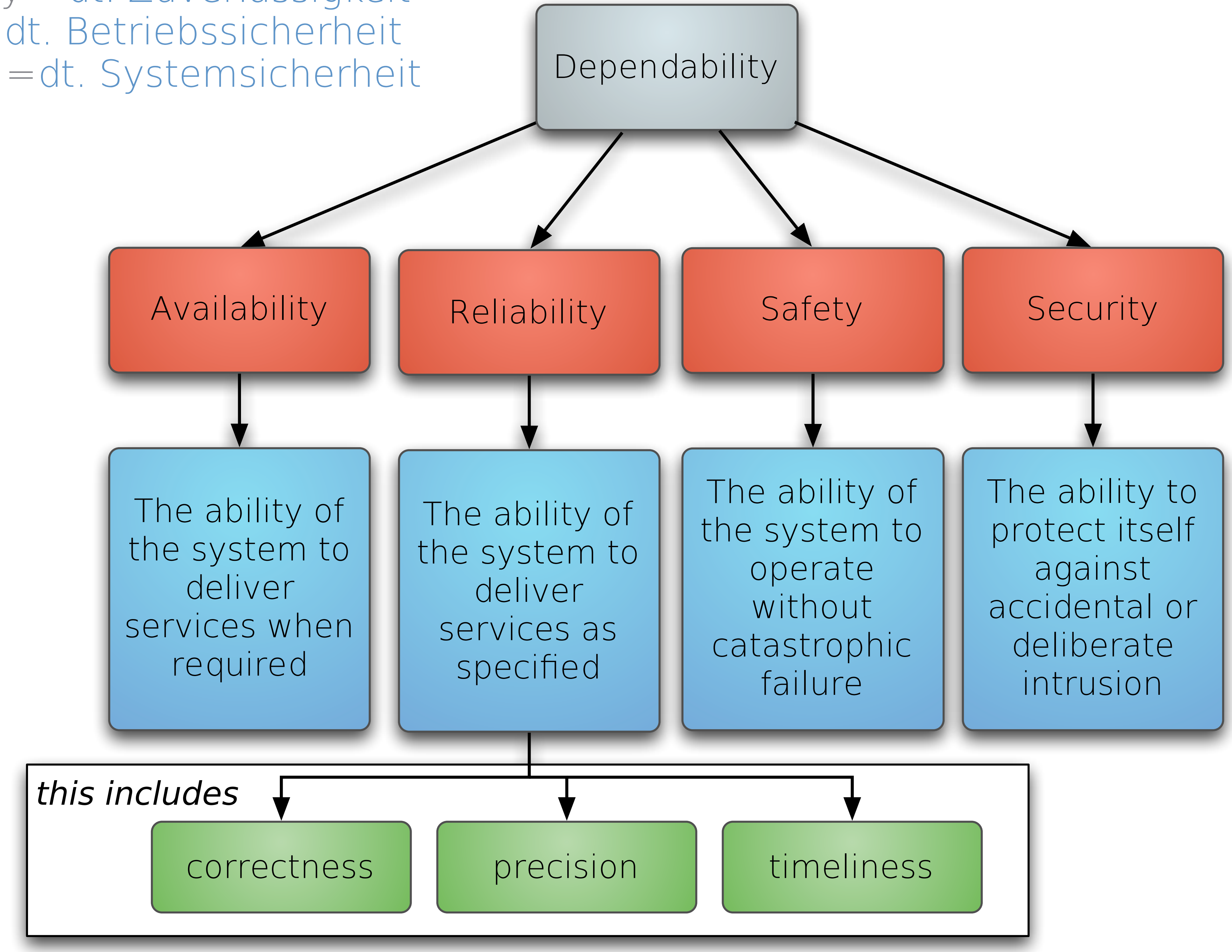
- **Good Software**
  Ian Sommerville; Software Engineering - Eighth Edition; Addison Wesley, 2007

Attributes of “good Software”

- **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


Dependability
Availability
Reliability
Safety
Security
The ability of the system to deliver services when required
The ability of the system to deliver services as specified
The ability of the system to operate without catastrophic failure
The ability to protect itself against accidental or deliberate intrusion
this includes
correctness
precision
timeliness

Software Quality

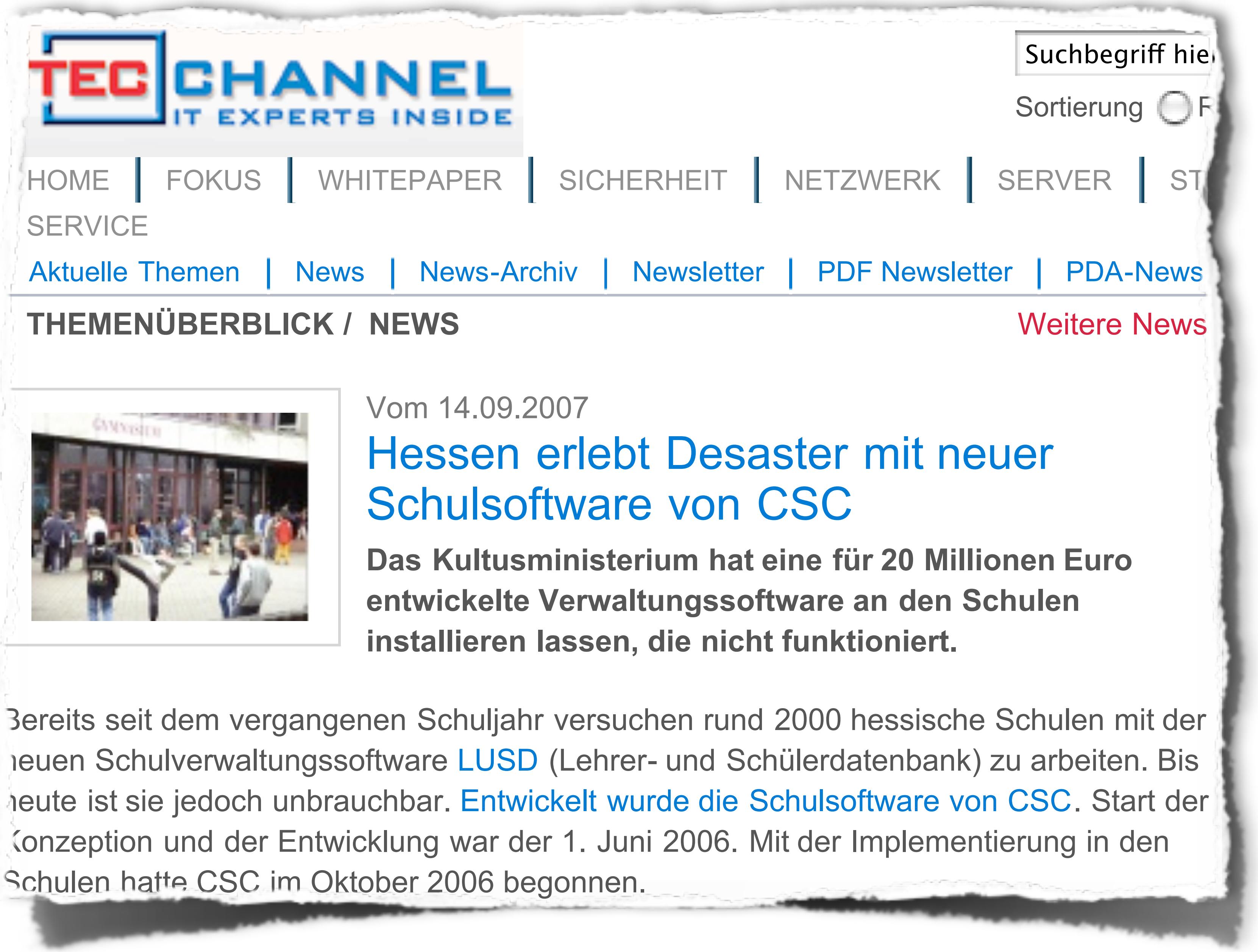
- … or the lack thereof.

Software failures can be disastrous.

- **Therac-25**
  People died due to an overdosis 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)
- . . .


TEC CHANNEL
IT EXPERTS INSIDE
Suchbegriff hie
Sortierung ○ F
HOME | FOKUS | WHITEPAPER | SICHERHEIT | NETZWERK | SERVER | ST
SERVICE
Aktuelle Themen | News | News-Archiv | Newsletter | PDF Newsletter | PDA-News
THEMENÜBERBLICK / NEWS
Weitere News

Vom 14.09.2007

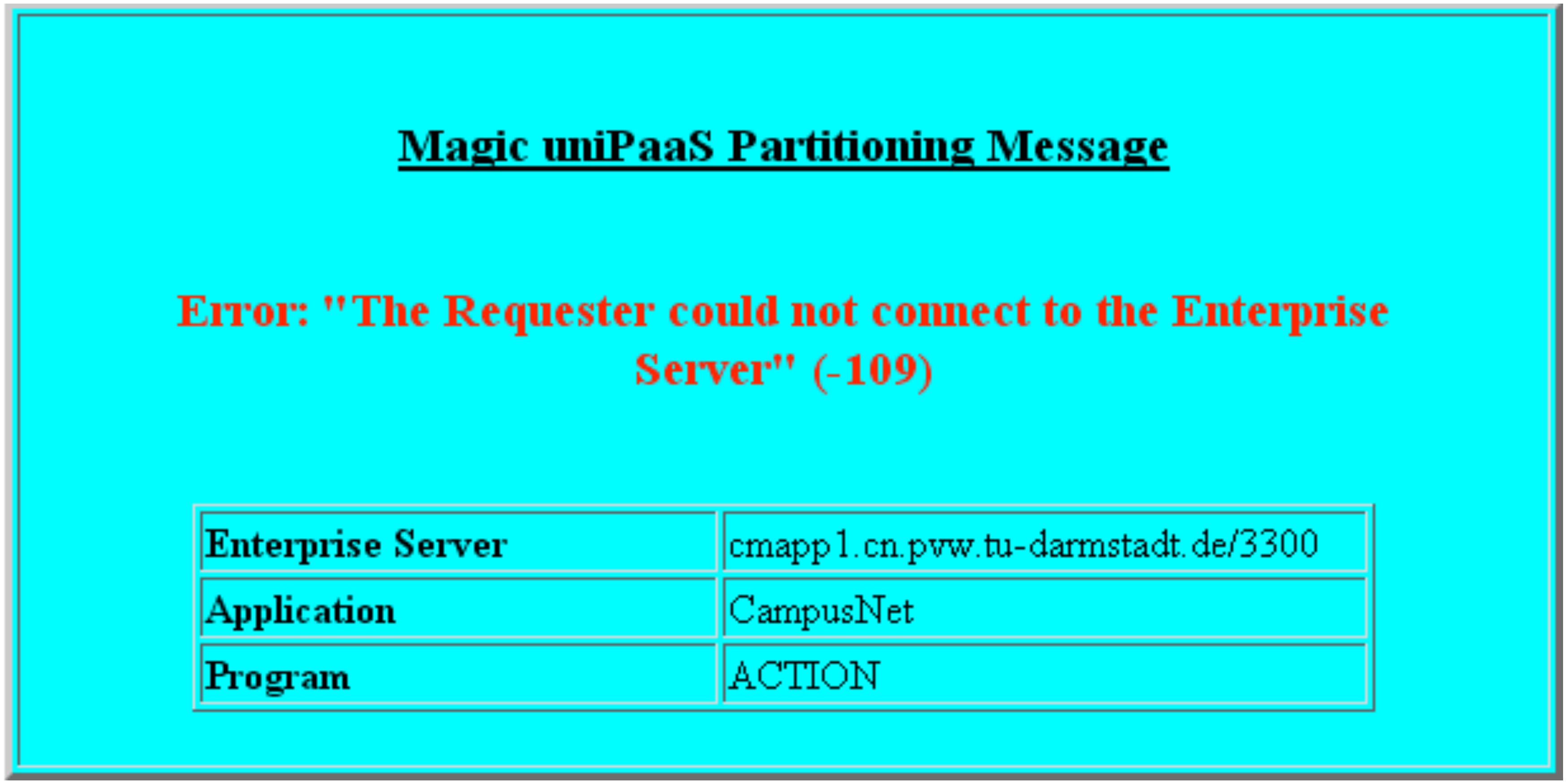
Hessen erlebt Desaster mit neuer Schulsoftware von CSC

Das Kultusministerium hat eine für 20 Millionen Euro entwickelte Verwaltungssoftware an den Schulen installieren lassen, die nicht funktioniert.

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.

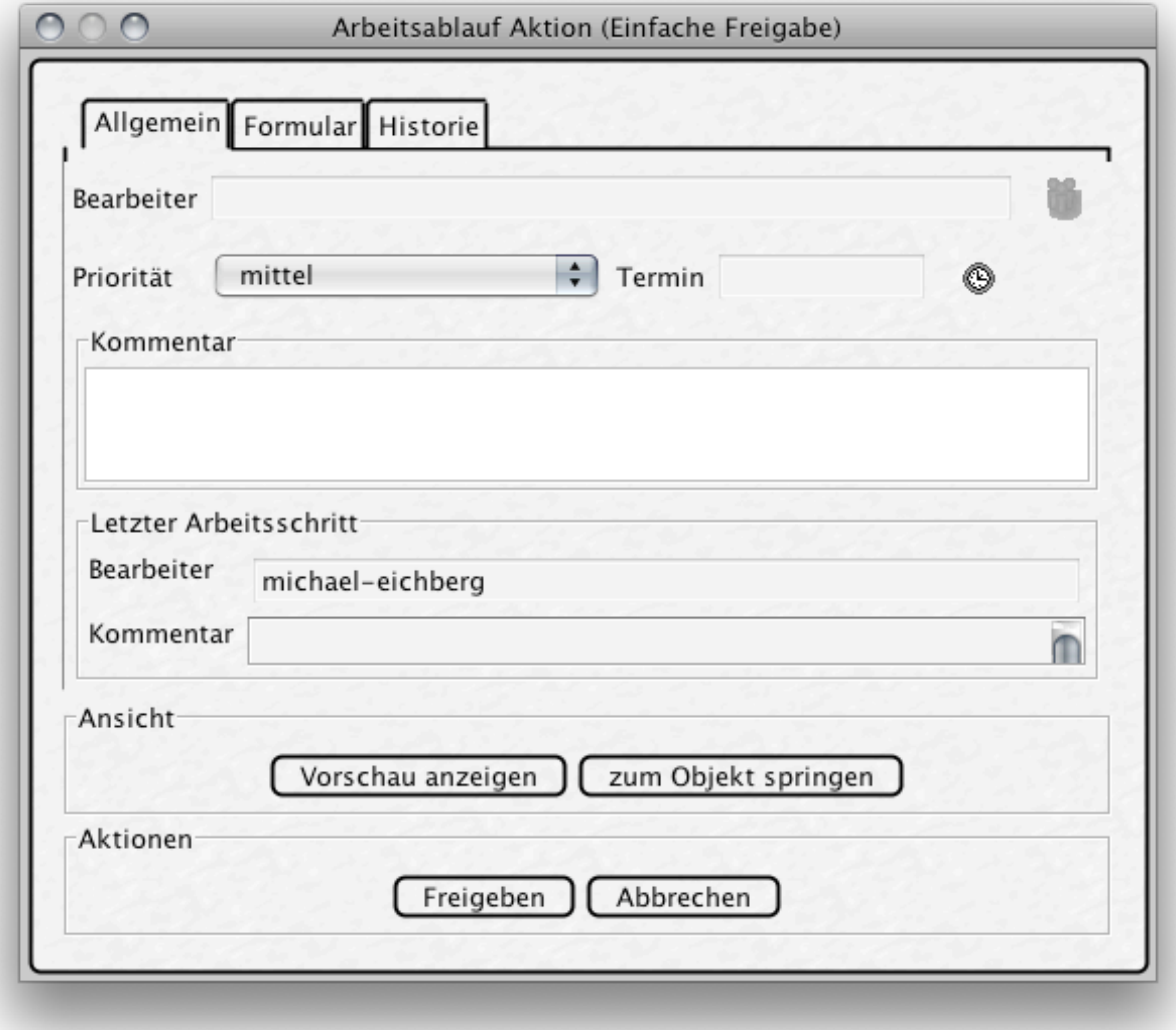
- CampusNet error message shown to the end user (2010)



Magic uniPaaS Partitioning Message

Error: "The Requester could not connect to the Enterprise Server" (-109)

Enterprise Server	cmapp1.cn.pvw.tu-darmstadt.de/3300
Application	CampusNet
Program	ACTION



Arbeitsablauf Aktion (Einfache Freigabe)

Bearbeiter

Priorität: mittel

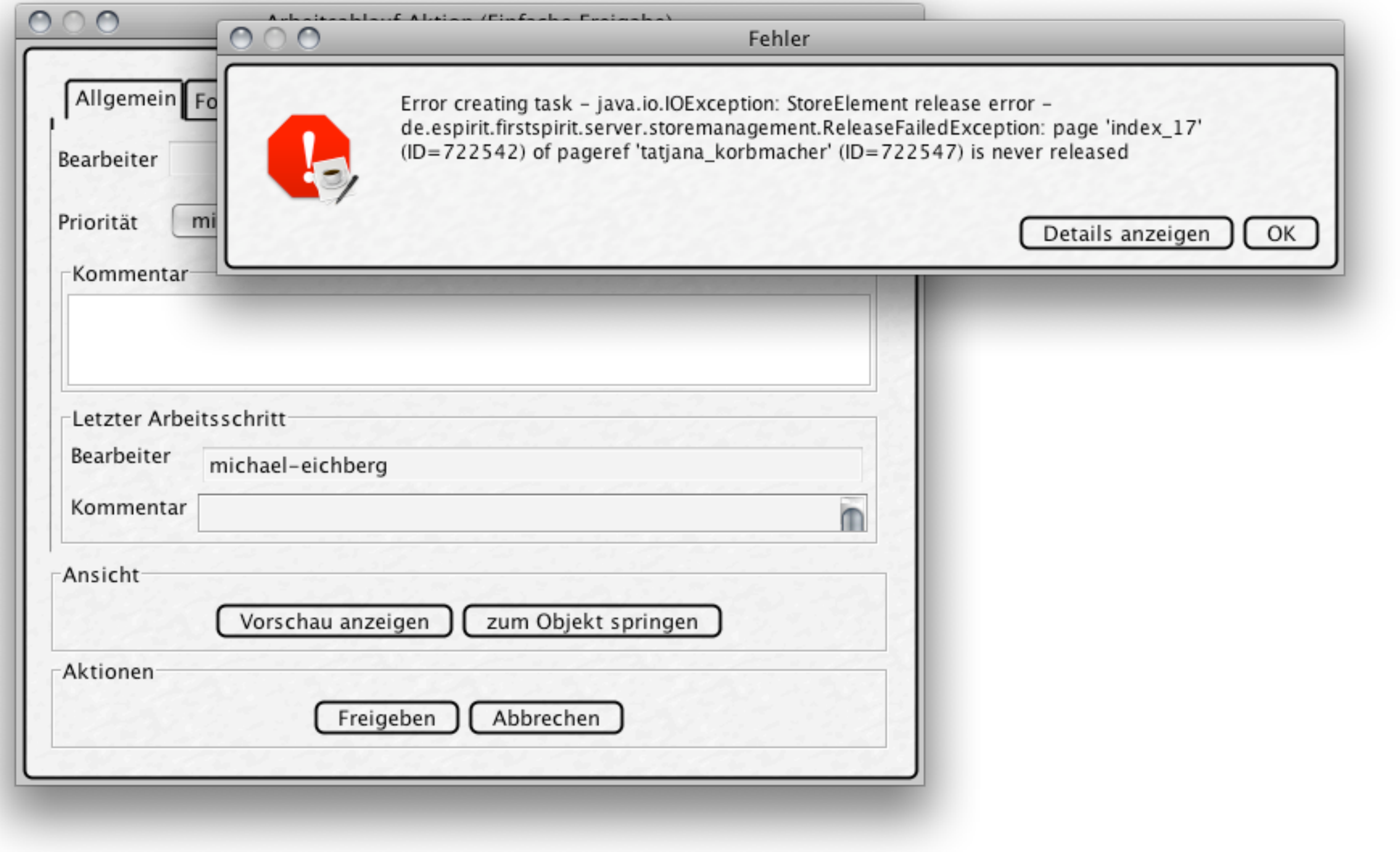
Termin

Kommentar

Letzter Arbeitsschritt

Bearbeiter: michael-eichberg

Kommentar



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

Bearbeiter

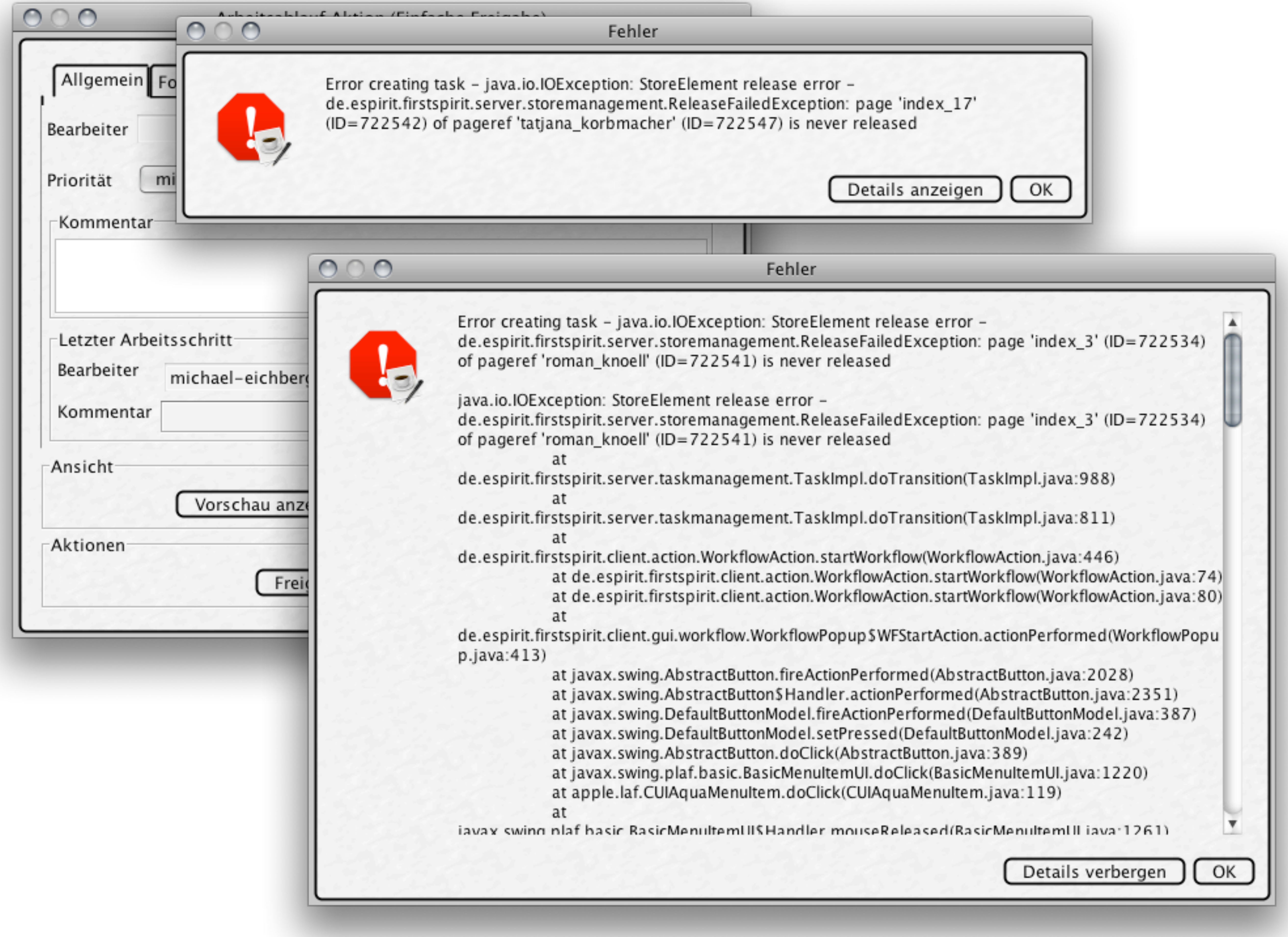
Priorität: mi

Kommentar

Letzter Arbeitsschritt

Bearbeiter: michael-eichberg

Kommentar



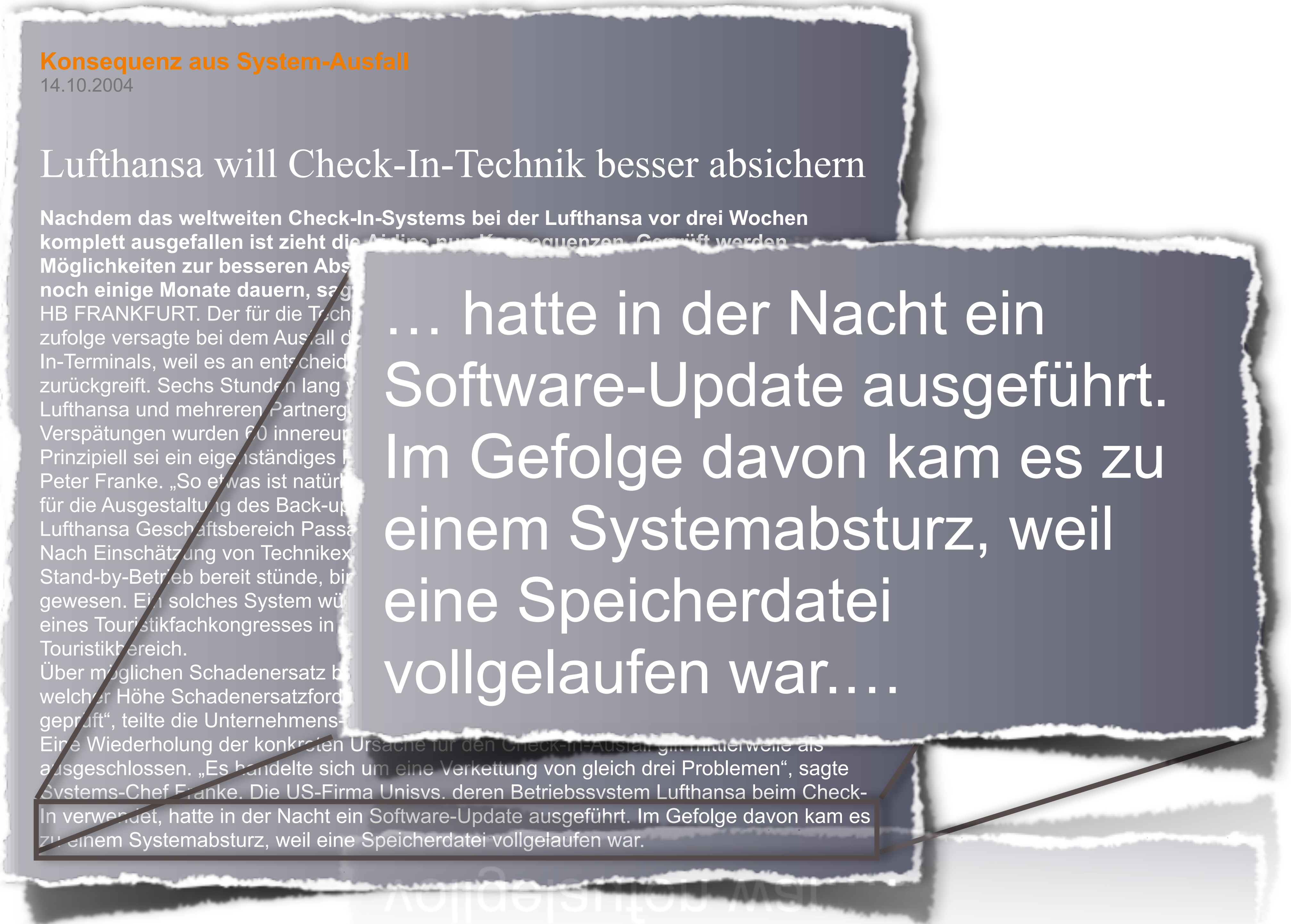
```
Error creating task – java.io.IOException: StoreElement release error –
de.espirit.firstspirit.server.storemanagement.ReleaseFailedException: page 'index_17'
(ID=722542) of pageref 'tatjana_korbmacher' (ID=722547) is never released
```

```
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)
```

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
Möglichkeiten zur besseren Abs
noch einige Monate dauern, sag**

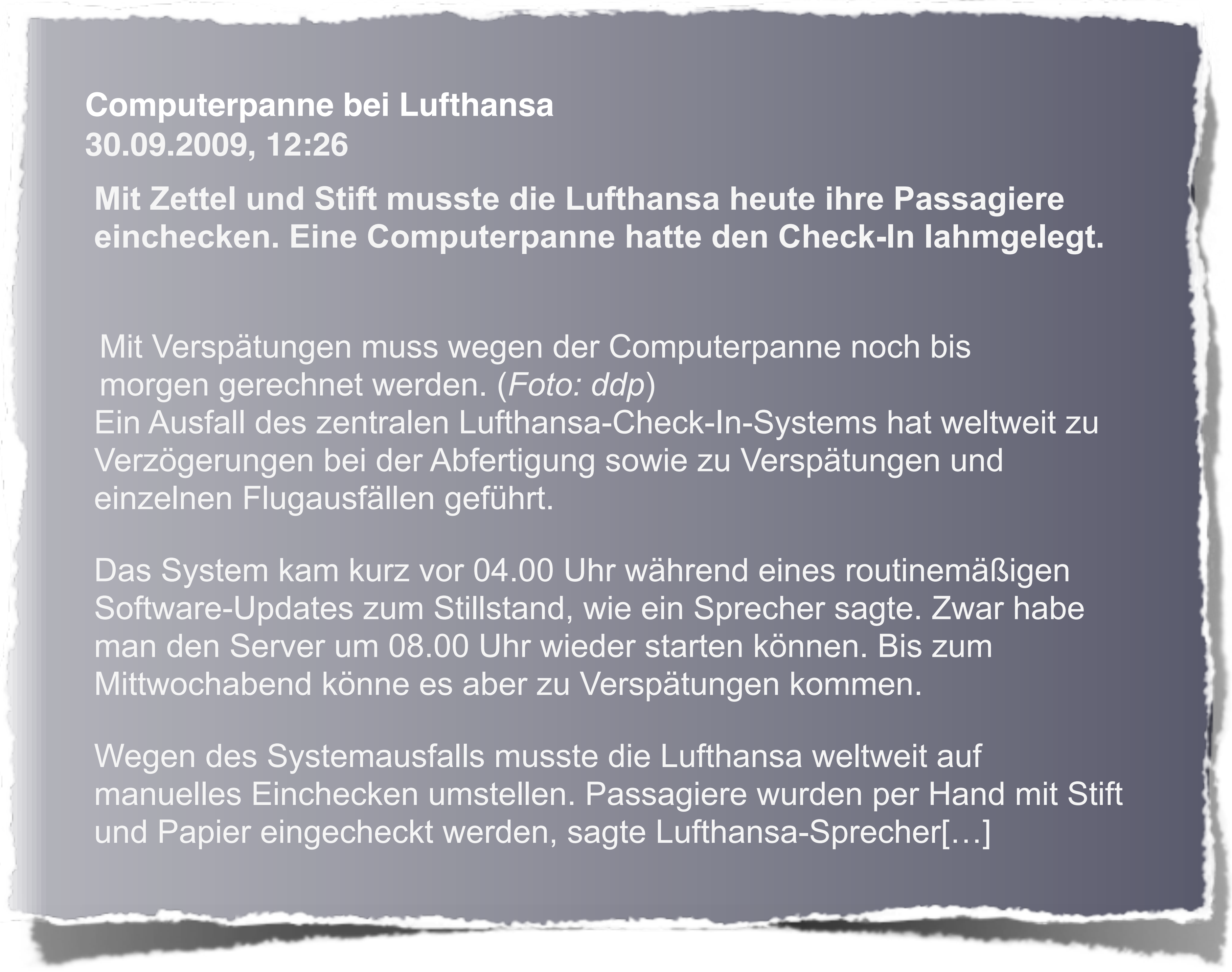
HB FRANKFURT. Der für die Tech
zufolge versagte bei dem Ausfall d
In-Terminals, weil es an entscheid
zurückgreift. Sechs Stunden lang v
Lufthansa und mehreren Partnerg
Verspätungen wurden 60 innereur
Prinzipiell sei ein eigenständiges
Peter Franke. „So etwas ist natürl
für die Ausgestaltung des Back-up
Lufthansa Geschäftsbereich Passa
Nach Einschätzung von Technikex
Stand-by-Betrieb bereit stünde, bi
gewesen. Ein solches System wü
eines Touristikfachkongresses in
Touristikbereich.
Über möglichen Schadenersatz b
welcher Höhe Schadenersatzford
geprüft“, teilte die Unternehmens-
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.…

Missing software quality in commercial software.

- 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*)

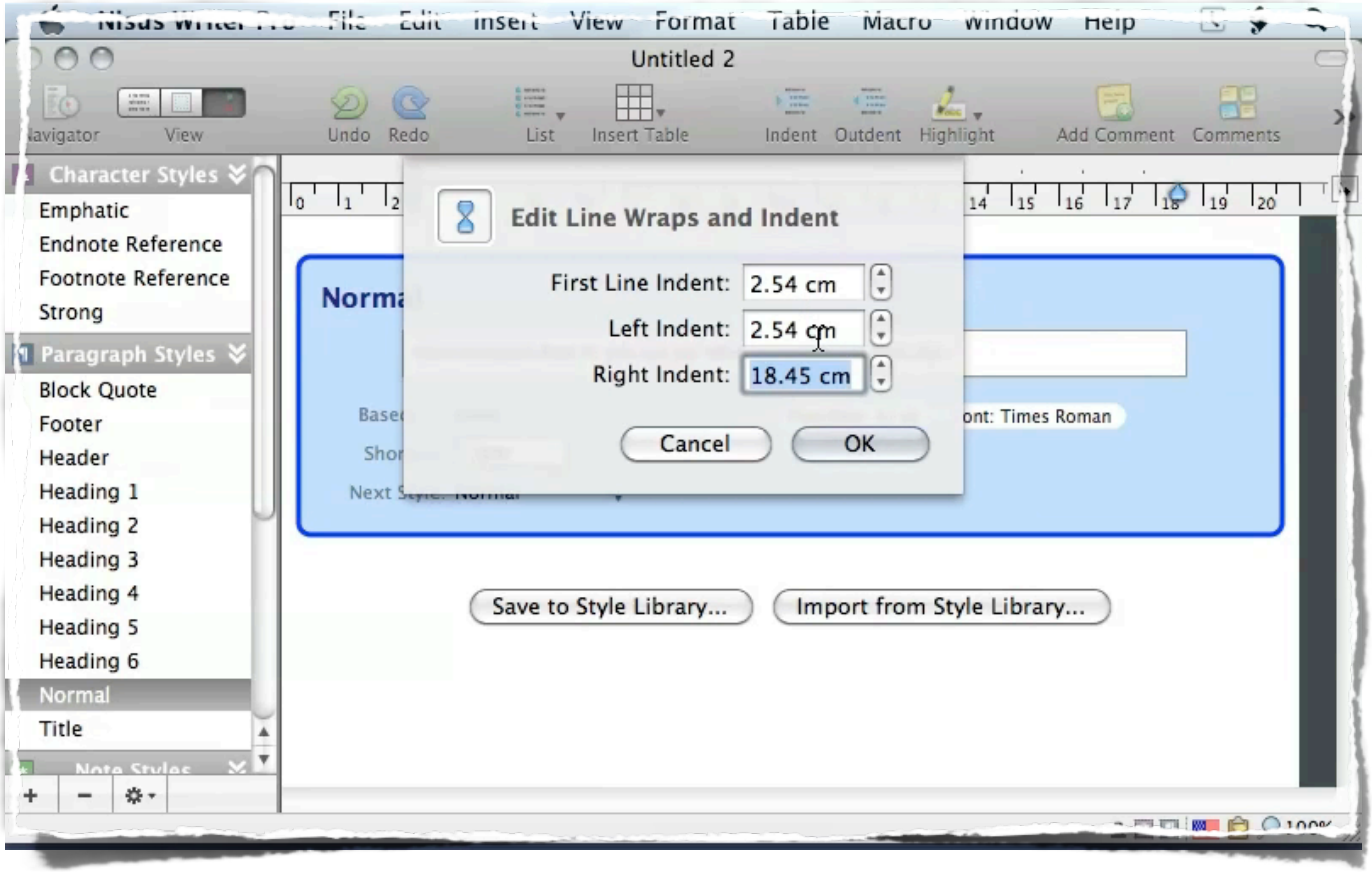
Ein Ausfall des zentralen Lufthansa-Check-In-Systems hat weltweit zu Verzögerungen bei der Abfertigung sowie zu Verspätungen und einzelnen Flugausfällen geführt.

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 eingecheckt werden, sagte Lufthansa-Sprecher[…]

Missing software quality in commercial software.

Nisus Writer Pro 1.4 - 2010



Edit Line Wraps and Indent

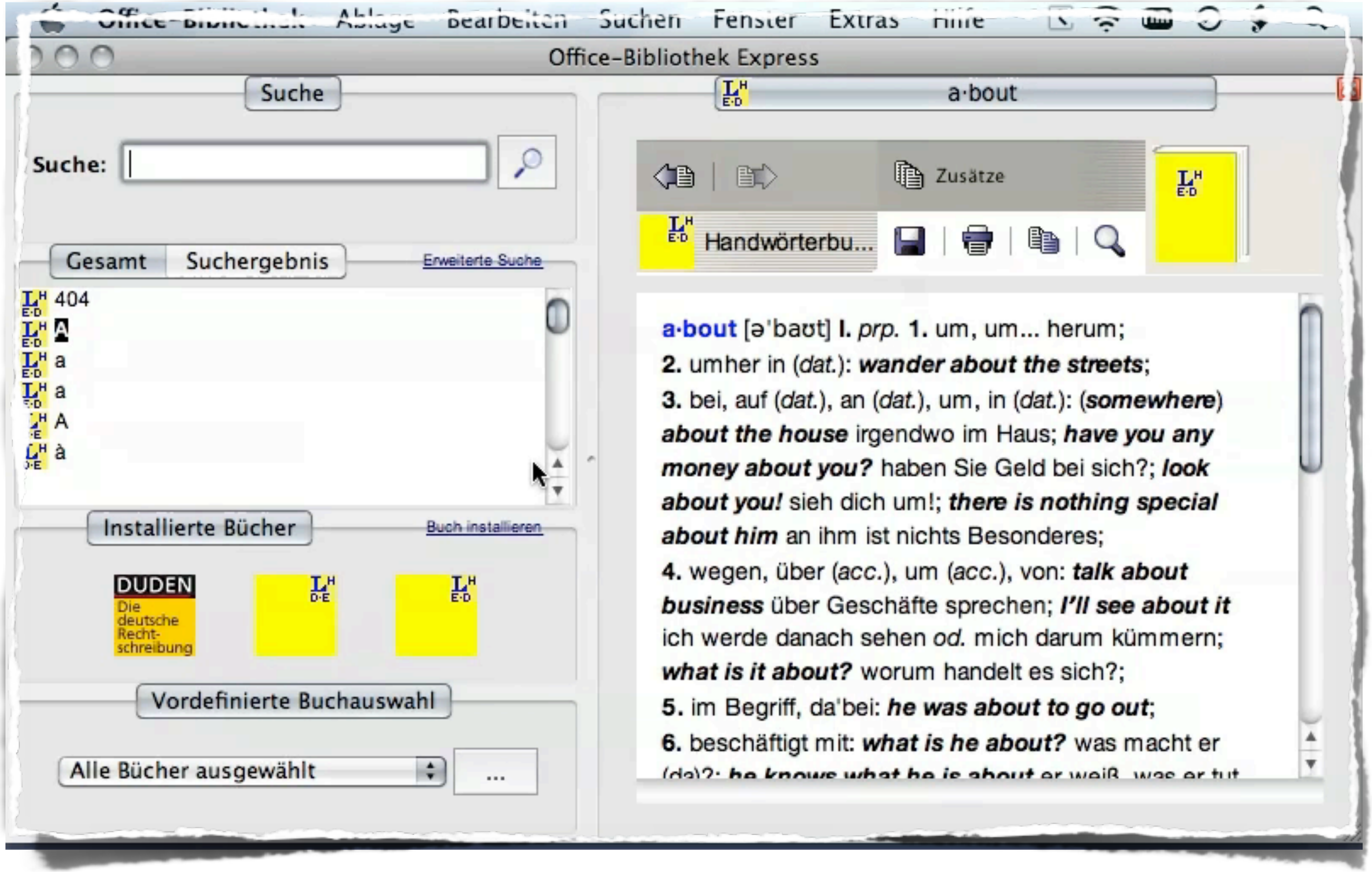
First Line Indent: 2.54 cm

Left Indent: 2.54 cm

Right Indent: 18.45 cm

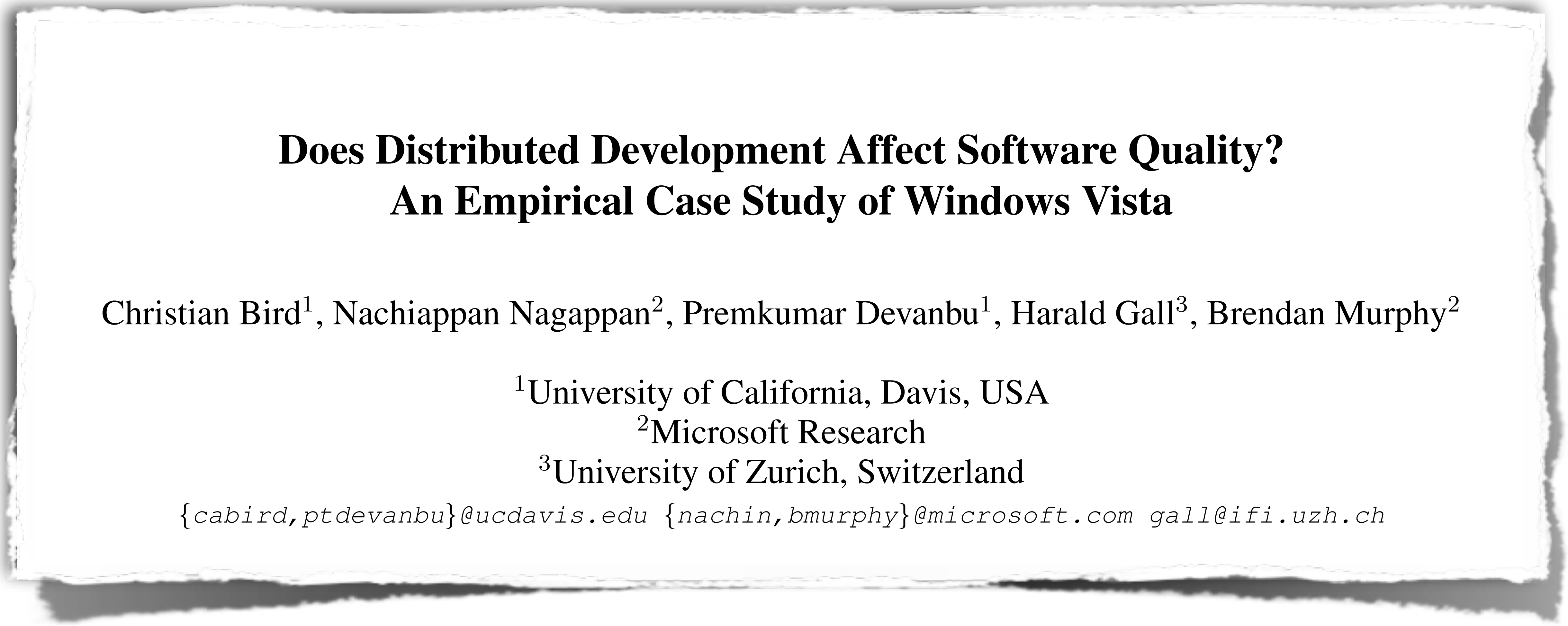
Missing software quality in commercial software.

Duden Office Bibliothek - 2010



a·bout [əˈbaʊt] **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

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

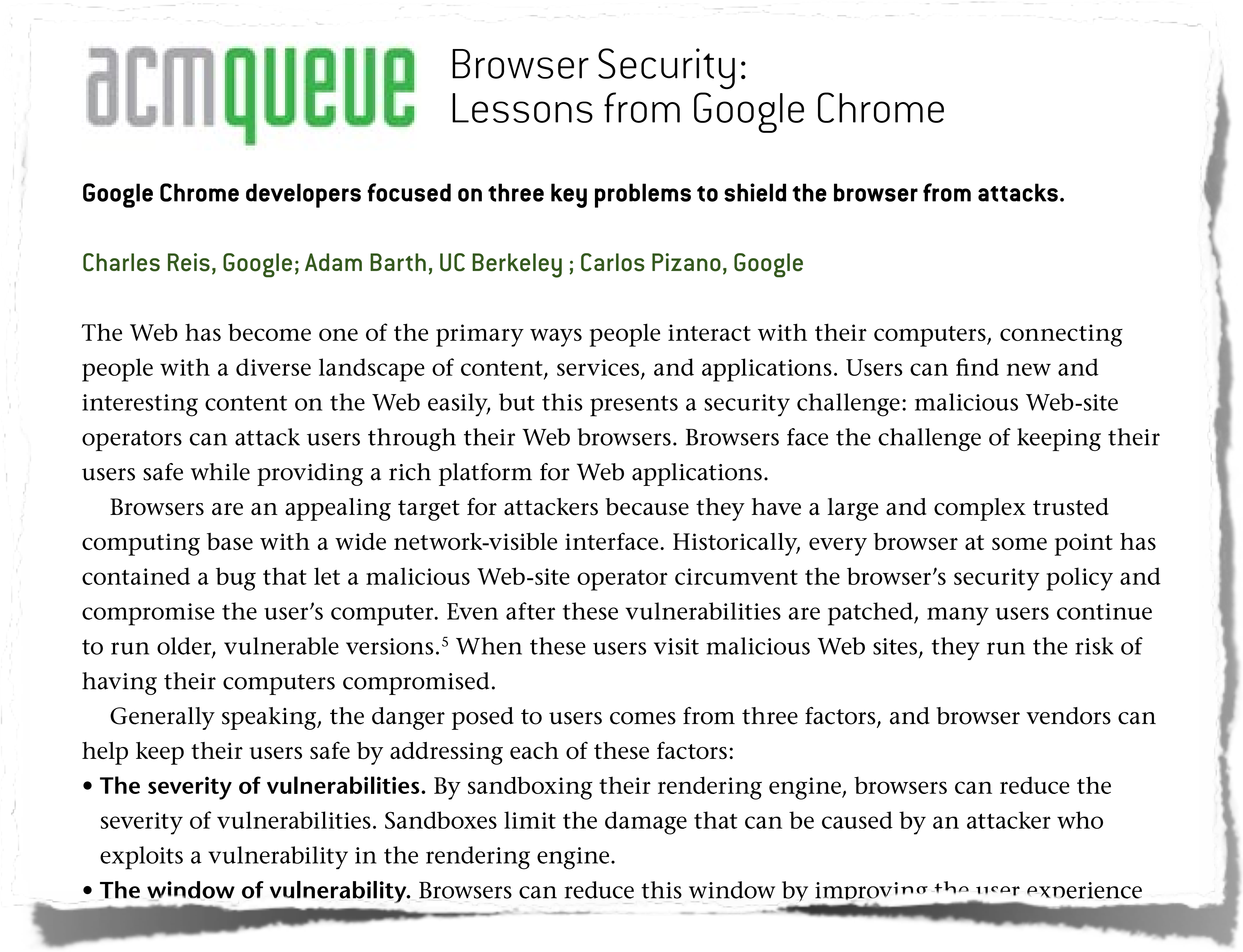


Does Distributed Development Affect Software Quality? An Empirical Case Study of Windows Vista

Christian Bird[1], Nachiappan Nagappan[2], Premkumar Devanbu[1], Harald Gall[3], Brendan Murphy[2]

[1]University of California, Davis, USA
[2]Microsoft Research
[3]University 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



acmqueue

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

Software Quality

- Summary

TECHNISCHE
UNIVERSITÄT
DARMSTADT

Recall the “fifteen principles of Software Engineering”.

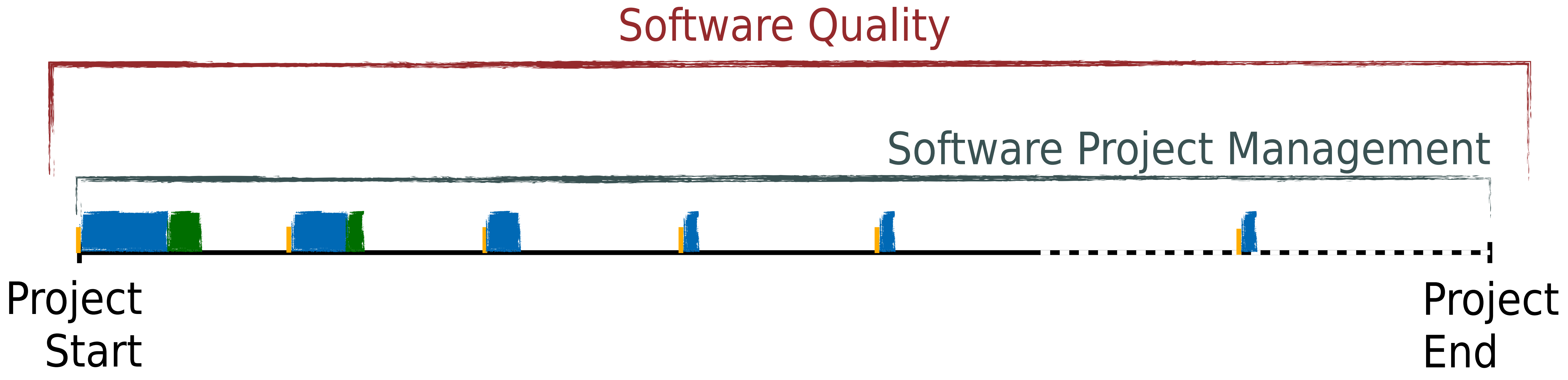
Take responsibility!

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.



Software Quality

Software Project Management

Project Start

Project End

🟦 Requirements Management

🟩 Domain Modeling