

Das gemeinsame Forschen, Lernen und Lehren von Studierenden und Lehrenden am Department für Informatik der Carl von Ossietzky Universität ist von gegenseitigem Respekt, Vertrauen und Toleranz geprägt. Dieses drückt sich insbesondere dadurch aus, dass wir:

1. unsere Mitverantwortung in der Gesellschaft, in der wir leben, aktiv wahrnehmen und die digitale Gesellschaft bewusst zum Wohle Aller gestalten.
2. die Herausforderungen der vernetzten Welt in friedlicher, internationaler Zusammenarbeit bearbeiten, jede Form von Nationalismus entschieden ablehnen und uns aktiv nationalistischen Tendenzen entgegenstellen.
3. uns dafür einsetzen, dass Menschen nicht wegen ihres Geschlechts, Hautfarbe, Sprache, Religion, politischer oder sonstiger Anschauung, nationaler oder sozialer Herkunft, der Zugehörigkeit zu einer nationalen Minderheit, des Vermögens, der Geburt oder eines sonstigen Status oder ihrer sexuellen Orientierung diskriminiert, benachteiligt oder diffamiert werden.
4. die tatsächliche Umsetzung der Gleichberechtigung der Menschen fördern und auf die Beseitigung bestehender Nachteile hinwirken.
5. nicht bereit sind, Diskussionen zu tolerieren, die faktenfrei, auf der Grundlage falscher Fakten, oder auf der Basis von Ressentiments geführt werden.

# Was tun bei Diskriminierung?

Ignorieren und verdrängen Sie Diskriminierung nicht, sondern setzen Sie sich zur Wehr:

- Schaffen Sie klare Verhältnisse und signalisieren Sie, dass dieses Verhalten unerwünscht ist.
- Dokumentieren Sie Diskriminierung schriftlich (insbesondere bei sexueller Belästigung).
- Sprechen Sie mit Menschen Ihres Vertrauens, mit der Fachschaft, mit verantwortlichen Dozentinnen und Dozenten sowie auch mit der zuständigen Beratungsstelle über diese negativen Erlebnisse.
- Erstanlaufstelle (Asta):  
<https://asta-oldenburg.de/erstanlaufstelle-diskriminierung/>
- Beratungsstelle conTakt:  
<https://uol.de/kontakt-beratungsstelle>

# Softwaretechnik

## Wintersemester 2019/2020

### inf005 - 2.01.005

Andreas Winter

<mailto:winter@se.uni-oldenburg.de>

Johannes Meier

<mailto:meier@se.uni-oldenburg.de>

# Zielgruppe

## Pflicht:

- Bachelor Informatik
- Bachelor Wirtschaftsinformatik
- Zwei-Fach-Bachelor Informatik
- Informatik (Master of Education (Wirtschaftspädagogik))

## Wahlpflicht/Professionalisierung:

- Bachelor Mathematik
- Bachelor Umweltmodellierung

gemäß aktueller fachspezifischer Anlage

# Vorlesung

## Dozent:

- Andreas Winter

## Termine:

- Donnerstags, 08:30-10:00, A14 1-102 (Hörsaal 2)
- Freitags, 10:15-11:45, A14 1-102 (Hörsaal 2)

## Terminplanung – Vorlesung (3/1)

- bis 10. Januar 2020:  
Vorlesung: 4-stündig; Donnerstags und Freitags

## Beginn:

- 17. Oktober 2019

## Raumänderung:

- 24. Oktober 2019 (A11 0-009)

## Keine Vorlesungen

- 01. November 2019 (Brückentag nach Reformationstag)

## Fragestunde vor der Klausur

- 31. Januar 2020

# Informationen zur Veranstaltung

## Folien zur Vorlesung

- (fast) alle gezeigten Folien i.d.R. nach der Vorlesung
  - in Stup.ID zur Vorlesung

## weitere Informationen zur Vorlesung

- Wiki
  - im Stup.ID zur Vorlesung
  - Hilfen und Informationen
- Forum
  - in Stup.ID zur Vorlesung
  - bitte zur allgemeinen Diskussion nutzen
  - **Lösungsideen dürfen gepostet und diskutiert werden**

alle Folien, Hilfen,  
etc. werden  
ausschließlich im  
Stup.ID zur  
**Vorlesung** bereit  
gestellt

## Tutorien

- Gruppen: 11
- Teilnehmer: bis zu 24 Teilnehmer pro Gruppe
- Beginn: 21./22./23. Oktober 2019
- Terminplanung: einstündig, Montags/Dienstags/Mittwochs

## Bemerkung

- zusätzliche „Großübungen“ zum Vorlesungstermin

Anmeldung und  
Priorisierung zu den Übungsgruppen  
bis **Fr. 18.10.2019, 12:00** über Stup.ID

# Tutorien - Montags

## Tutorium E (0)

Tutor: Kimberly Hebig  
[kimberly.hebig@uol.de](mailto:kimberly.hebig@uol.de)

Termin: Montags, 18:00-19:00

Raum: A3 4-402

## Tutorium F (0)

Tutor: Kimberly Hebig  
[kimberly.hebig@uol.de](mailto:kimberly.hebig@uol.de)

Termin: Montags, 19:00-20:00

Raum: A3 4-402



# Tutorien - Dienstags

## Tutorium L (0)

Tutor: Johannes Meier  
[meier@se.uol.de](mailto:meier@se.uol.de)

Termin: Dienstags, 09:00-10:00

Raum: A3 4-402

## Tutorium I (0)

Tutor: Kimberly Hebig  
[kimberly.hebig@uol.de](mailto:kimberly.hebig@uol.de)

Termin: Dienstags, 10:00-11:00

Raum: A3 4-403

## Tutorium G (0)

Tutor: Fabian Skok  
[fabian.skok@uol.de](mailto:fabian.skok@uol.de)

Termin: Dienstags, 18:00-19:00

Raum: A5 1-160

## Tutorium J (0)

Tutor: Kimberly Hebig  
[kimberly.hebig@uol.de](mailto:kimberly.hebig@uol.de)

Termin: Dienstags, 11:00-12:00

Raum: A3 4-403

## Tutorium H (0)

Tutor: Jan Hofmann  
[jan.hofmann@uol.de](mailto:jan.hofmann@uol.de)

Termin: Dienstags, 19:00-20:00

Raum: A5 1-160

# Tutorien - Mittwochs

## Tutorium A (0)

Tutor: Lukas Kilian

[lukas.leander.kilian@uol.de](mailto:lukas.leander.kilian@uol.de)

Termin: Mittwochs, 12:00-13:00

Raum: A3 4-402

## Tutorium C (0)

Tutor: Florian Hustede

[florian.hustede@uol.de](mailto:florian.hustede@uol.de)

Termin: Mittwochs, 16:00-17:00

Raum: A3 4-402

## Tutorium B (0)

Tutor: Lukas Kilian

[lukas.leander.kilian@uol.de](mailto:lukas.leander.kilian@uol.de)

Termin: Mittwochs, 13:00-14:00

Raum: A3 4-402

## Tutorium D (0)

Tutor: Florian Hustede

[florian.hustede@uol.de](mailto:florian.hustede@uol.de)

Termin: Mittwochs, 17:00-18:00

Raum: A3 4-402

## Verteilung auf Übungsgruppen durch Losverfahren

- Anmeldung zur Vorlesung „Softwaretechnik 1“ in Stup.ID
- Sammlung von Terminwünschen zu den Tutorien bis zum **Freitag, 18. Oktober 2019, 12:00** in Stup.ID
- Verlosung der Übungsgruppen Freitag Nachmittag
- *Schlafmützen* melden sich bei Johannes Meier

clever  
priorisieren

## Bildung der Kleingruppen (4 Personen)

- **nach Verteilung auf Übungsgruppen über Stup.ID**
- konkrete Übungsaufgaben werden nur im Tutorium der Kleingruppe besprochen und zurückgegeben
- *Schlafmützen* klären mit Tutor Zuordnung zur Kleingruppe

## Losverfahren und Softwaretechnik

- „das neue Losverfahren ist nicht toll durchdacht und der **Algorithmus schlecht programmiert**. Ich habe schon jetzt zwei Freunde, die sich für 5 Termine bei einem Tutorium "beworben" haben und diese haben keinen Platz dort erhalten. Der angebliche Koeffizient, der mit der ersten verpassten Teilnahme erstellt wird für diese Person, damit diese dann eine erhöhte Chance bekommt, scheint wohl nicht zu existieren. Das vorherige System war deutlich besser, da hat man sich kurz darüber aufgeregt, dass der Uni-Server so lahm ist, aber man kam in sein gewünschtes Tutorium rein. Bitte überdenken Sie das Verfahren noch einmal gründlich!“

Ist die Qualität der Software zur Verlosung der Übungsplätze schlecht?

Was ist überhaupt „Software-Qualität“?

## Aufgabenblätter

- Bearbeitung in Kleingruppen (4 Personen)
- Bearbeitungszeit: eine Woche, Fr. bis Fr. 10:00
- erstes Aufgabenblatt: 18.10.2019

## Hinweise

- Aufgabenblätter bauen teilweise aufeinander auf
- fast alle Aufgaben waren einmal Klausuraufgaben
- Softwaretechnik ist Teamwork und erfordert Diskussionen
- Teamwork heißt nicht
  - wenige arbeiten, der Rest sieht zu !!!
- Es gibt *keine* Musterlösungen
  - Softwaretechnik muss man machen, das lernt man nicht durch Lesen von Musterlösungen
  - individuelle Lösungen dürfen im Forum diskutiert werden
- Korrektur der Aufgabenblätter
  - nur Fehler werden angemerkt, es gibt keine (Bonus-)Punkte

SVN-Einführung  
in den Übungen

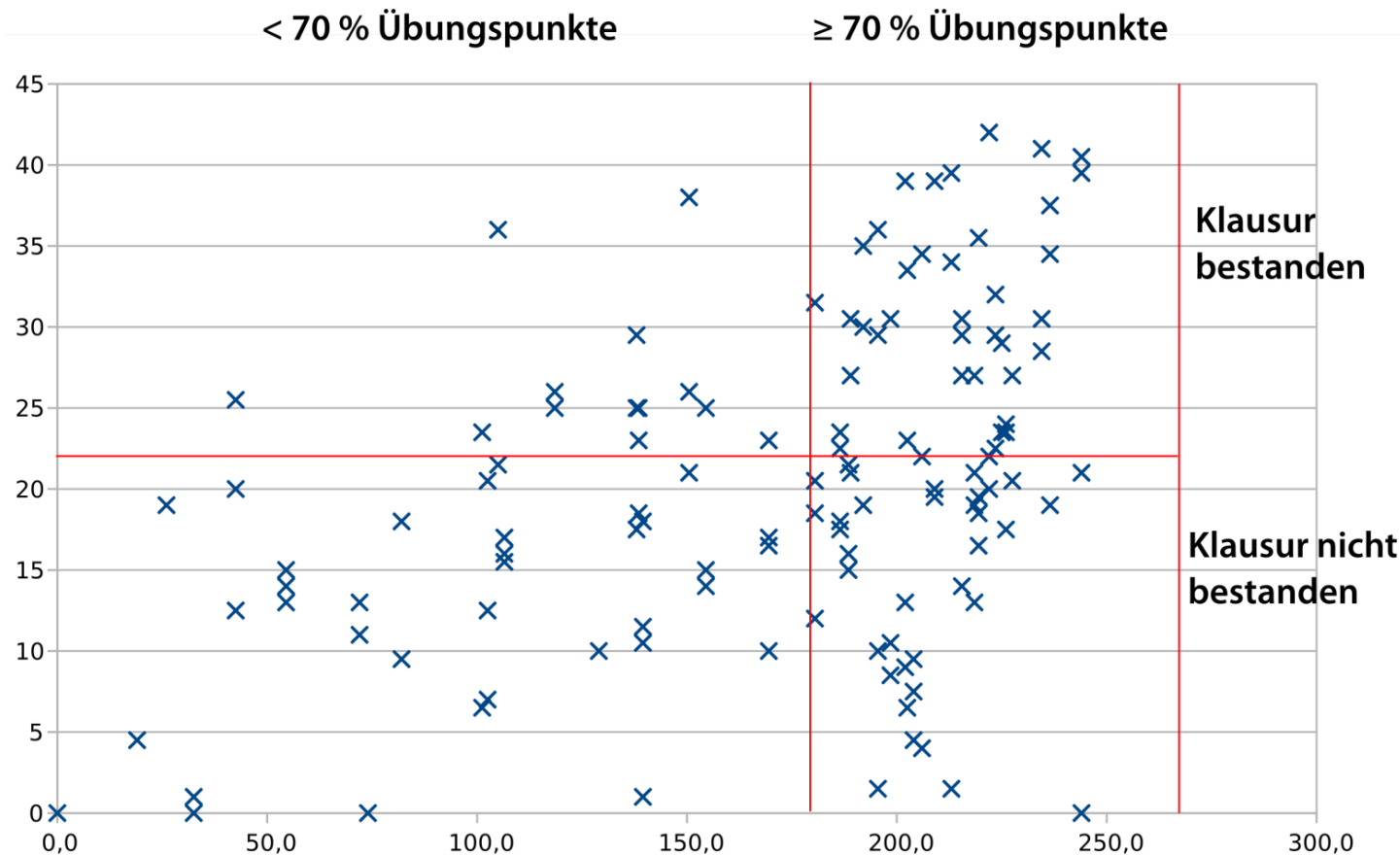
## Freitags (Vorlesung)

- Ausgabe Themenblatt i
  - in Stup.ID zur Vorlesung
- Rückgabe Themenblatt i-1 (bis 10:00)
  - über das jeweilige Arbeitsgruppen-SVN (Gruppe XX)
    - Lösungen zu Themenblatt YY ins Unterverzeichnis assignments einstellen
    - für Graphiken und Texte: eine PDF-Datei: st\_gXX\_aYY.pdf einstellen
    - für Programme: Java-Dateien (.java) in UTF-8 Codierung einstellen
    - auf alle Seiten Namen der Bearbeiter und Übungsgruppe schreiben
    - ausschließlich die Lösung in assignments einchecken

## Montags/Dienstags/Mittwochs (Übung)

- Vorbesprechung zu Themenblatt i
- Präsentation und Besprechung der Lösungen zu Themenblatt i-1

# Übungsteilnahme und Klausurergebnis



## Folgerung:

- Bearbeitung der Übungsaufgaben erhöht die Wahrscheinlichkeit die Klausur zu bestehen
- Das gilt aber nur für aktive Bearbeitung der Übungsaufgaben im Team

Softwaretechnikklausur Wintersemester 2014/15

## Softwaretechnik

- erlernt man nicht durch Ansehen von Bildchen
- erlernt man nur durch selbständiges Anwenden und Diskutieren der präsentierten Methoden und Techniken
- muss man aktiv machen



[<http://fullnomad.com/wp-content/uploads/2015/08/make-it-so.png>]



## Termin

- Klausur (120 min)
  - Dienstag, 04.02.2020, 14:30-15:30 Uhr, A14 1-101/2 (Hörsaal 1/2)
- Wiederholungsklausur (120 min)
  - Montag, 06.04.2020, 14:30-16:30 Uhr, A11 1-101 (Hörsaal B)

## Teilnahmebedingungen

- rechtzeitige Anmeldung zur Klausur
  - Anmeldung zur Klausur möglich bis zum 28.01.2020 (MEZ)
  - Abwesenheit bei der Klausur trotz Anmeldung gilt als Fehlversuch
- Leistungsnachweis (6 KP-Punkte)
- Erreichen von 45% der Gesamtpunkte der Klausur

# Termine (Zusammenfassung)

17.10.2019

- Vorlesungsbeginn

18.10.2019

- Ausgabe 1. Aufgabenblatt

18.10.2019

- Priorisierung der Tutorien

21./22./23.10.2019

- Übungsbeginn

28.01.2020

- Anmeldung zur Klausur

31.01.2020

- Fragestunde vor der Klausur

04.02.2020

- Klausur

30.03.2020

- Anmeldung zur  
Wiederholungsklausur

06.04.2020

- Wiederholungsklausur

## Erasmus+

- Teilstipendium für  $\geq 3$  Monate
  - je nach Zielland (2018): 300/360/420 €/Monat
- keine Studiengebühren im Ausland
- erleichterte Anerkennung von Leistungen
- direkte Ansprechpartner

## Bewerbung

- über jeweiliges Studienfach
- Sprachvoraussetzungen beachten

## Kontakt

- <https://uol.de/erasmus-bewerbung/>
- Prof. Dr. Martin Fraenzle



## Erasmus+ Partnerschaften in der Informatik

Land	Universität	Niveau
	University of Vaasa	BSc/MSc
	Institut Supérieur d'Action Internationale et de Production, Angers (*)	BSc/MSc
	Universite de Nice Sophia Antipolis	BSc/MSc
	Edge Hill University, Ormskirk (*)	BSc/MSc
	Università Politecnica delle Marche, Ancona (*)	MSc
	Mykolas Romeris University, Vilnius (*)	BSc
	University of Oslo (*)	BSc/MSc
	Universitatea Transilvania din Brasov (*)	BSc/MSc
	Politehnica University of Bucharest (*)	BSc/MSc
	Universidad de Cádiz	BSc/MSc
	Universidad Rey Juan Carlos, Madrid	BSc/MSc



# Was ist Softwaretechnik?

# Softwaretechnik != Programmierung

# 1.1 Zielsetzung der Vorlesung



## Problembewusstsein schaffen

- Ursachen für Probleme der Informationstechnik erkennen
- Notwendigkeit für Softwaretechnik erkennen

## Überblick vermitteln

- Terminologie der Softwaretechnik verstehen und anwenden
- zentrale Tätigkeiten zur Erstellung großer Softwaresysteme erkennen und einordnen
- zentrale Rollen der bei der Erstellung großer Softwaresysteme beteiligten Stakeholder erkennen und einordnen

## Handwerkzeug vermitteln

- Vorgehensweisen der Softwaretechnik bewerten und anwenden
- Prinzipien, Methoden und Techniken der Softwaretechnik bewerten und anwenden

## Grundbegriffe der Objekt-orientierten Software-Entwicklung

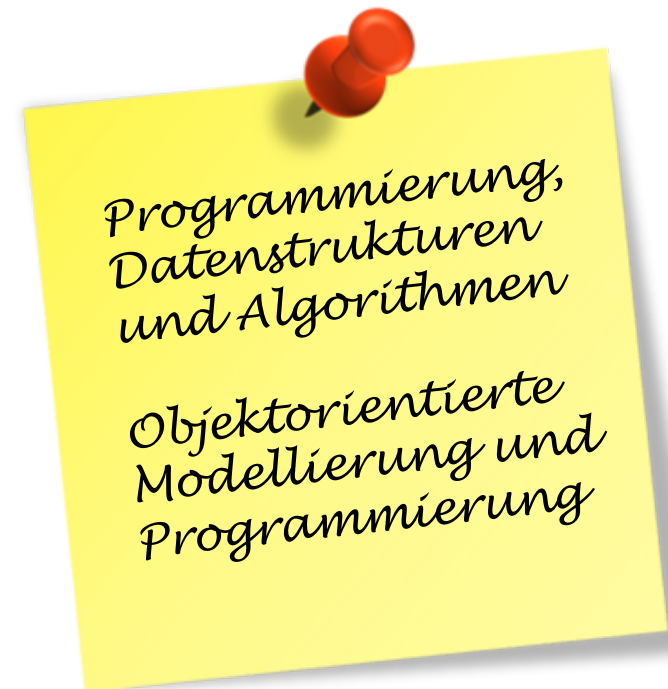
- Klasse, Objekt
- Variable, Methode
- Zuweisung, Kontrollstrukturen (Sequenz, Verzweigung, Iteration)
- Vererbung

## Grundlegende Datenstrukturen

- Liste, Menge, Multimenge, Stapel ...

## Grundlegende Algorithmen

- Suchverfahren
- Sortierverfahren



- hier:  
mehrere Mitarbeiter  
über einen langen Zeitraum
- hier nicht:  
ein "Freizeit"-Programmierer

Erstellung eines großen Softwaresystems  
für einen Auftraggeber  
im Rahmen eines Projekts

- hier:  
Individualsoftware
- hier nicht:  
Standardsoftware  
(COTS, commercial  
of the shelf)

- hier:  
einmalige Zielverfolgung
- hier (eher) nicht:  
dauerhafte Weiterentwicklung

## Software

- ist eine Menge von Programmen oder Daten zusammen mit begleitenden Dokumenten, die für ihre Anwendung notwendig oder hilfreich sind.

## Projekt

- ist die zeitlich begrenzte, einmalige Verfolgung eines vorgegebene Ziels, die zu einem definierten Ergebnis führt.

## Produkt

- ist ein i. Allg. für einen Auftraggeber erstelltes, in sich abgeschlossenes Ergebnis eines erfolgreich durchgeführten Projekts.

[Hesse et al., 1984] [Balzert, 2001, S. 23ff]

## Aktivitäten der Software-Entwicklung

## Objektorientierte Modellierung mit UML

- Klassendiagramme

## Anforderungserhebung

- Vision
- Anforderungen
- Anforderungsdefinition

## dynamische Modellierung

- Sequenzdiagramme
- Aktivitätsdiagramme
- Zustandsdiagramme

## Software-Systementwurf

- Architektur
- Architekturmuster

## Qualitätssicherung

- Qualitätskriterien,
- Testen

## Software-Evolution

- Software-Wartung
- Reverse-Engineering
- Reengineering

# Nicht Thema dieser Vorlesung

- Kommunikationstechniken
- Präsentationstechniken
- Erhebungstechniken  
(Befragung, Interview, Gruppendiskussion)
- Semantik von Modellierungssprachen
- formale Methoden der Softwaretechnik
- Codierung  
(Programmieren, Algorithmik)
- Projektmanagement

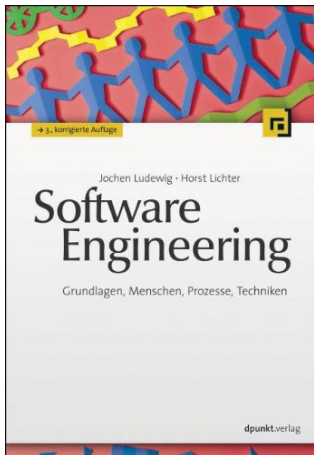
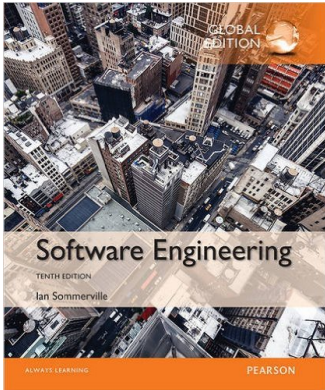
## Softwaretechnik Vorlesung    Software-Projekt

- Überblick über Methoden und Techniken zum Software-entwicklung
    - ST-Grundlagen und Prinzipien
    - iterative und agile Vorgehensmodelle
    - UML Grundlagen
    - Anforderungserhebung
    - Software-Architektur und Entwurfsmuster
    - Software-Qualität
    - Software-Evolution
  - Tools
    - SVN
    - UML Designer
- Anwendung von Methoden und Techniken zur Softwareentwicklung und Programmierung
    - konkrete Entwicklung
    - (a kind of) SCRUM
    - Modellierung und Programmierung
    - User-Stories
    - Model-View-Presenter
    - JUnit
  - Tools
    - GIT
    - Visual Paradigm
    - Atlassian Stack (JIRA, Confluence)

## 1.2 Literatur



# Literatur



- Ian Sommerville: Software Engineering, Addison-Wesley Longman, Amsterdam, 10. Auflage (Global Edition). 2015.
- Jochen Ludewig, Horst Lichter: Software Engineering, Grundlagen, Menschen, Prozesse, Techniken, dpunkt, 3. Auflage, 2013.
- Helmut Balzert: Lehrbuch der Softwaretechnik, Basiskonzepte und Requirements Engineering, Spektrum, Heidelberg, 3. Auflage, 2009.
- Helmut Balzert: Lehrbuch der Softwaretechnik: Software-Management, Spektrum, Heidelberg, 2. Auflage, 2008.
- Helmut Balzert: Lehrbuch der Softwaretechnik: Entwurf, Implementierung, Installation und Betrieb, Spektrum, Heidelberg; 3. Auflage, 2012.



## Zeitschriften

- IEEE Transactions on Software Engineering, IEEE Computer Society, (seit 1975).
- Software Engineering Notes, ACM Special Interest Group on Software Engineering, (seit 1976).
- Softwaretechnik-Trends, Gesellschaft für Informatik, Fachgruppen Softwaretechnik, Ada, Requirements-Engineering, Test, Analyse und Verifikation von Software, Objektorientierte Software Entwicklung, Software-Reengineering (seit 1980).
- IEEE Software, IEEE Computer Society, (seit 1984).
- ACM Transactions on Software Engineering and Methodology, Association for Computing Machinery, (seit 1992).



## 1.3 Software-Fehler

## Fehler

- Bei der rechnergestützten Auswertung der OB-Wahl in Neu-Ulm 1994 wurde zunächst eine Wahlbeteiligung von 104% festgestellt.

## Ursache

- In der Auswertungssoftware hatte sich ein Faktor 2 eingeschlichen.

## Folge

- Neuauszählung



[Partsch, 1998, S. 1]

# Software-Fehler

## Fehler (1.10.2017)

- Geldautomat der Volksbank Oldenburg frisst EC-Karte direkt nach Einschieben ohne lesbare Fehlermeldung



[https://upload.wikimedia.org/wikipedia/commons/5/57/Hauptstelle\\_Oldenburg\\_Fassade\\_Lange\\_Straße.jpg](https://upload.wikimedia.org/wikipedia/commons/5/57/Hauptstelle_Oldenburg_Fassade_Lange_Straße.jpg)

## Ursache

- „Betreiber hat vor 14 Tagen ein Software-Update aufgespielt; Karten, die mit Chip nach oben eingegeben werden, werden als fehlerhaft erkannt und eingezogen“

## Folge

- Kein Bier auf dem Kramermarkt



<https://bankenverband.de/media/uploads/2015/06/04/sicher-mit-karte.jpg>



## Fehler

- vollständiger Batterieausfall in Fahrzeugen eines deutschen Automobil-Herstellers

## Ursache

- Fehler in der Software-gesteuerten Innenbeleuchtung

## Folge

- Rückrufaktion mit einem (geschätzten) Schaden von mehreren Millionen DEM



[Partsch, 1998, S. 1]

# Software-Fehler

## Fehler (2.8.2007)

- Online-Übermittlung eines Papers zur Veröffentlichung in einem Konferenzband wird trotz korrekter Eingabe nicht akzeptiert



## Ursache

- Software-Entwickler hat am Tag zuvor "etwas" geändert und war dann in Urlaub gefahren.

## Folge

- manuelle Übermittlung aller Beiträge per Mail.

"Our programmer changed something simple yesterday for the front matter that is causing this problem, and he's out for a personal reason today."  
(aus der Fehlermail)

# Software-Fehler

## Fehler:

- Netzwerk stürzte bei Xerox mehrfach komplett ab.

## Ursache:

- Inkompatibilität zwischen Microsoft Windows XP (beta) und Cisco 5000 Routers.

## Folge:

- Warnung an alle 50.000 U.S. Mitarbeiter von Xerox "not to install XP betas without permission or they'd face disciplinary action".



[ACM SIGSOFT Software Engineering Notes, vol. 26, no 4, (2001), S.6]



# Software-Fehler

## Fehler:

- Ausfall der Aktienhandels-Website von Charles Schwab & Co am 24.2.1999, 5 min nach Eröffnung des Handels an der NYSE.

## Ursache:

- Fehlgeschlagenes Software Upgrade in einem neuen Mainframe System in Phoenix.

## Folge:

- weitere Fehler in anderen Online-Aktienhandel-Systemen (E-Trade, Waterhouse, Ameritrade, Datek)



[ACM SIGSOFT Software Engineering  
Notes, vol. 24, no 3, (1999), S.25]

# Software-Fehler

## Fehler:

- Absturz von Air Inter Flug 148 (A 320) am 20.1.1992 bei Strasbourg.

## Ursache:

- schlechte Wetterbedingungen
- viermal zu hohe Sink-Geschwindigkeit (3,300 fpm) wurde von Crew nicht erkannt
  - wenig Kommunikation und nicht ausreichende Cross-Checks
  - Anzeige der Sink-Geschwindigkeit erlaubt 2 Modi
    - Sink-Geschwindigkeit (Vertical Speed Mode) 3,300 fpm (Ist)
    - Sink-Winkel (Flight Path Angle) 3.3' (Soll)



## Folge:

- 87 von 96 Passagieren/Crew starben

[<http://blog.martindoms.com/2011/01/24/poor-ui-design-can-kill/>,  
<http://aviation-safety.net/database/record.php?id=19920120-0>]

## Fehler

- USS Yorktown treibt während eines Manövers im September 1997 manövrierunfähig bei Cape Charles, VA.

## Ursache

- Fehler in einer Routine zum Abfangen einer Division durch Null in einer Windows NT Applikation.
- Die "Null" wurde als manuelle Eingabe einer enormen Datenmenge interpretiert.



## Folge

- "Atlantic Fleet officials said the ship was dead in water for about 2 hours and 45 minutes."

## Fehler:

- Patriot-Rakete verfehlt im 2. Golf-Krieg eine abzuschießende Scud-Rakete und schlägt in eine US-Kaserne in Dhahran ein.

## Ursache:

- Patriot war ursprünglich nicht als Anti-Raketen-Rakete konzipiert
- Steuercomputer lief 4 Tage statt maximal 14 Stunden ununterbrochen.
- internes 24 Bit Register lief über und verursachte Rundungsfehler bei der Bahnberechnung.
- Timerintervall von Manager von  $1/8$  sec auf  $1/10$  sec geändert. Ohne diese Änderung wäre der Überlauf nicht aufgetreten.



## Folge:

- 28 Tote

[ACM SIGSOFT Software Engineering  
Notes, vol. 16, no 3, (1991), S.19f]



## Fehler:

- Ariane 5 kommt auf ihrem Jungfernflug 501 vom Kurs ab.

## Ursache:

- Software zur Überwachung der Flugbahn (SRI) wurde teilweise aus Ariane 4 übernommen.
- SRI rechnete nach dem Start weiter, obwohl Ergebnisse nicht mehr benötigt wurden.
- Flugbahnberechnung erzeugte Überlauf wodurch wichtige Flugdaten überschrieben wurden.
- Das SRI-System und sein Backupsystem schalteten sich aufgrund des Fehlers ab.

## Folge:

- Aufruf der Selbstzerstörung der Rakete (Verlust: 500 000 000 USD)



[Pfleeger, 1998, S.37]

- typische Programmierfehler (z.B. Speicherüberläufe)
- fehlende Qualitätssicherung (z.B. Test nach Erweiterung)
- Inkompatibilitäten verschiedener Systemkomponenten
- schlechtes UI-Design
- inkorrekte/unvollständige Fehlerbehandlung
- Wiederverwendung von Code, der unter anderen Voraussetzungen entwickelt wurde, und deren Verwendungsvoraussetzungen unklar waren
- Eingriffe in Softwaresysteme durch unqualifizierte Mitarbeiter
- Fehlerhafte Nutzung