



Prof. h.c. Dr.

Frank J. Furrer

Antrittsvorlesung

Technische Universität Dresden

Lehrstuhl Softwaretechnologie

21. Januar 2016

Kurzlebenslauf

2015 (1. Juli): **Professor h.c.** der Fakultät für Informatik der Technischen Universität Dresden (TUD)



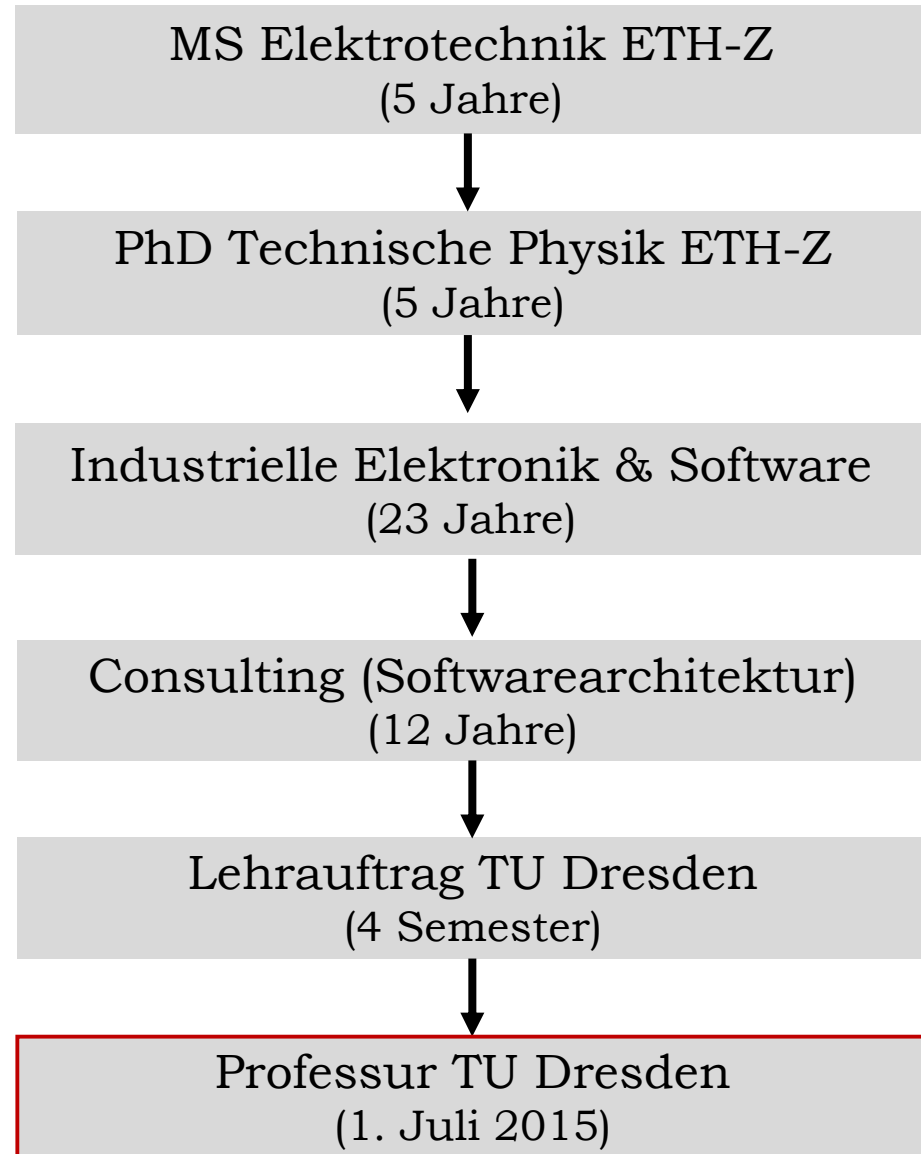
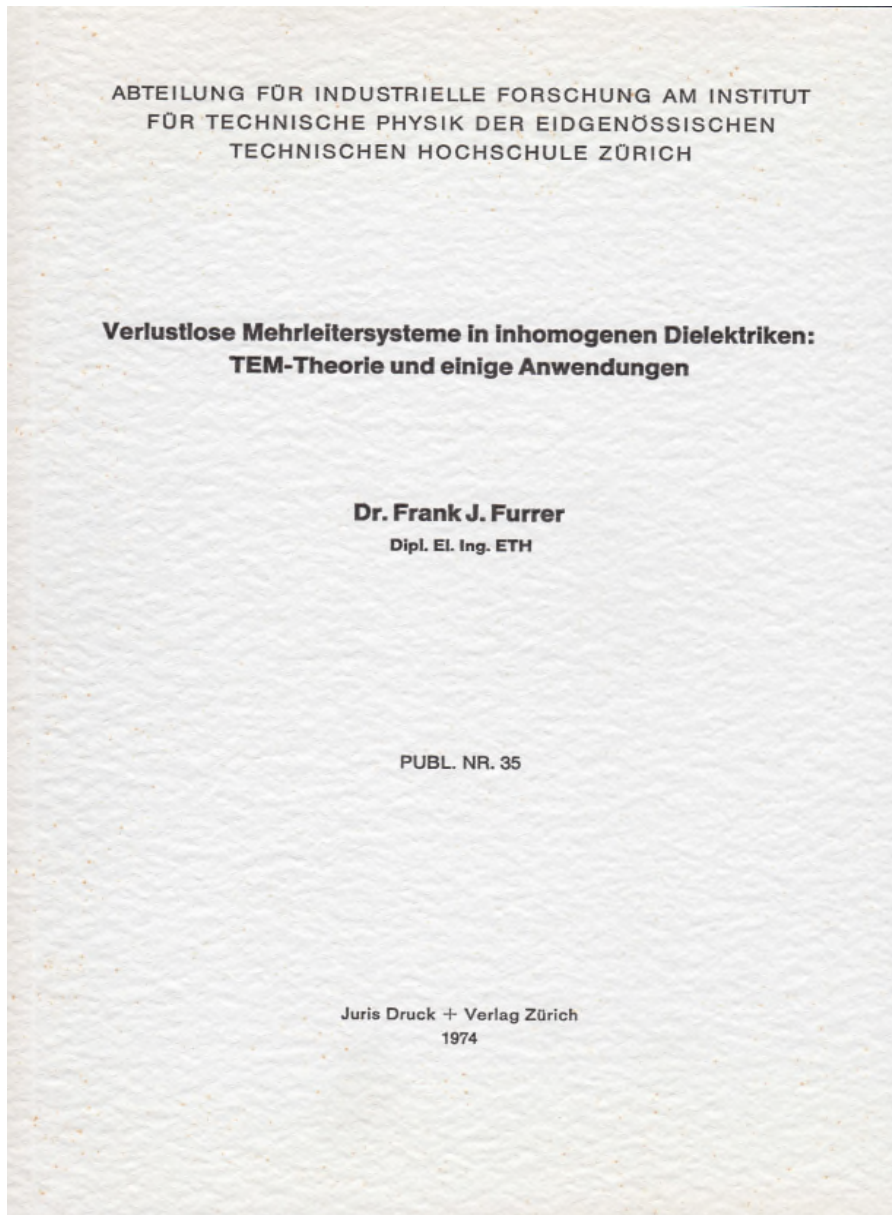
2013/14: **Lehrbeauftragter** TUD Dresden

1976 -2011: **Industrie-Laufbahn** im Gebiet der *industriellen Steuerungstechnik* und der *Architektur sehr grosser Softwaresysteme*

1975: **Ph.D.EE** (Dr. sc. techn. ETHZ) der Schweizerischen Technischen Hochschule Zürich (Insitut für Technische Physik)

1970: **MS** (Diplom) der Schweizerischen Technischen Hochschule Zürich, Fakultät für Elektrotechnik

1945 (27. Januar): Geboren in Zürich (Schweiz)



Frank J. Furrer

SOFTWARE:
Gratwanderungen zwischen
Erfolgsgeschichten und Katastrophen?

Was ist Software?

... eine kurze Einführung für Nicht-Informatiker



Software:

Liste von
Computerinstruktionen
(Computersprache)



Ausführender
Computer

Interaktion mit der Umwelt



<http://www.deluxevectors.com>

<http://www.rosacea.org>

<http://www.midnightcheese.com>



... das erste Programm

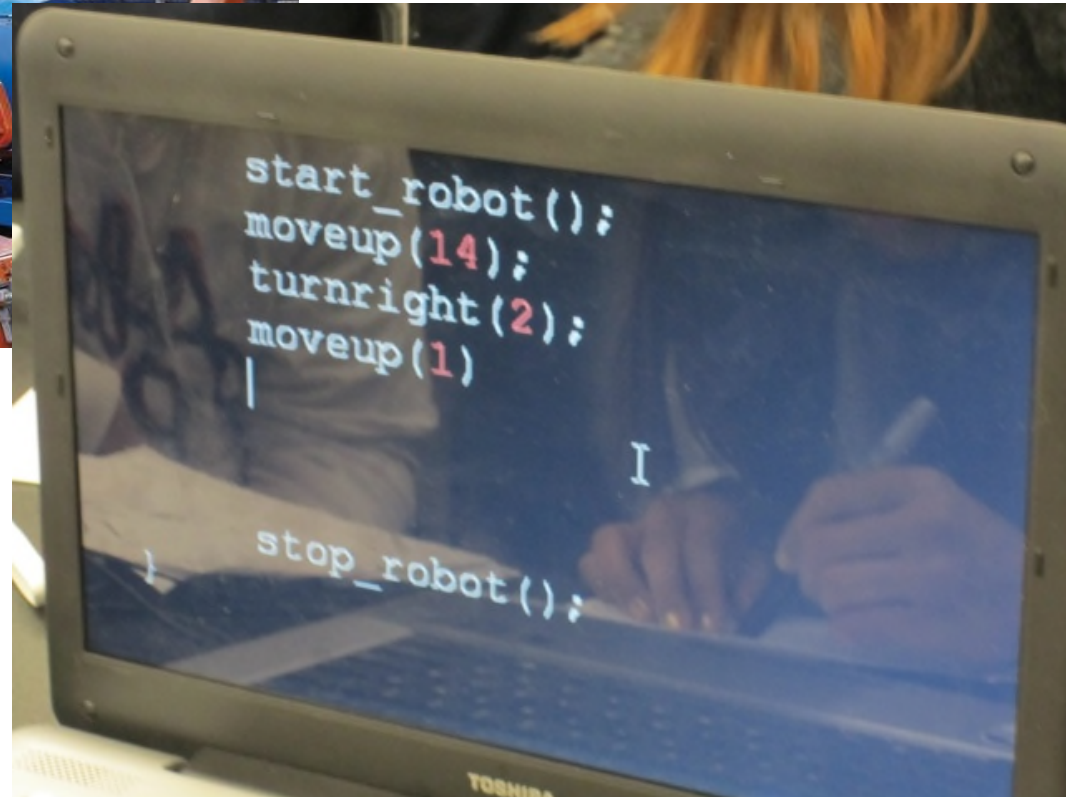
```
public class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("Hello World!");  
    }  
}
```

<http://www.noesispoint.com>

<http://www.brewerystudios.co.uk>



Roboterarm Programmierung



<http://www.projectvision.net>



«The Power of Software»

Die Kraft von Software ist (nahezu) allmächtig geworden

Hauptrolle: **Informatiker**



<http://www.cartoonaday.com>

Informatiker: Erschaffer des digitalen Universums

Wo findet man überall Software?

... noch eine kurze Einführung für Nicht-Informatiker

Software ist allgegenwärtig

<http://expertbeacon.com>



e-Commerce
(Internet-Shopping)

On-Line Banking
(Finanztransaktionen)



<https://www.zkb.ch>

<http://www.birminghammail.co.uk>



Strassen-
verkehrs
Steuerung

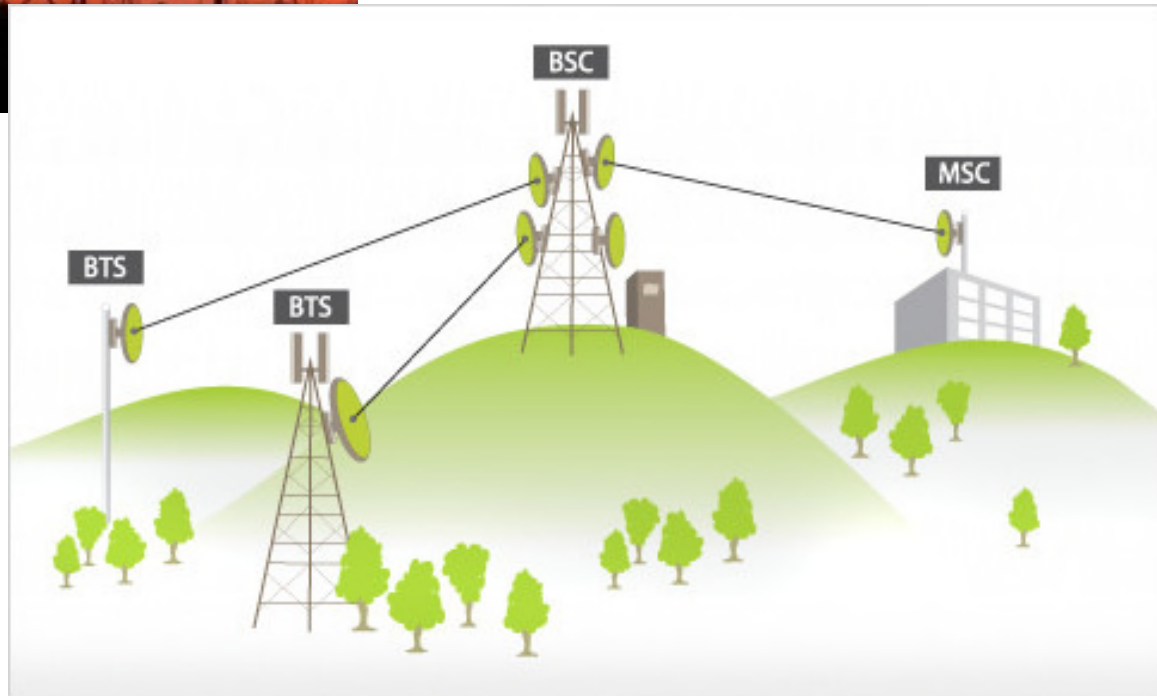
Luftverkehrs-
Überwachung



<https://www.linkedin.com>

Energieverteilung

<http://ecowatch.com>



<http://old.zycomm.co.uk>

Mobilnetzwerk

Computer-
gesteuertes
Fliegen

<http://www.airlinereporter.com>



<http://www.motorauthority.com>



Autonome Fahrzeuge

Chirurgische Roboter



Militärische Roboter

Herzschrittmacher



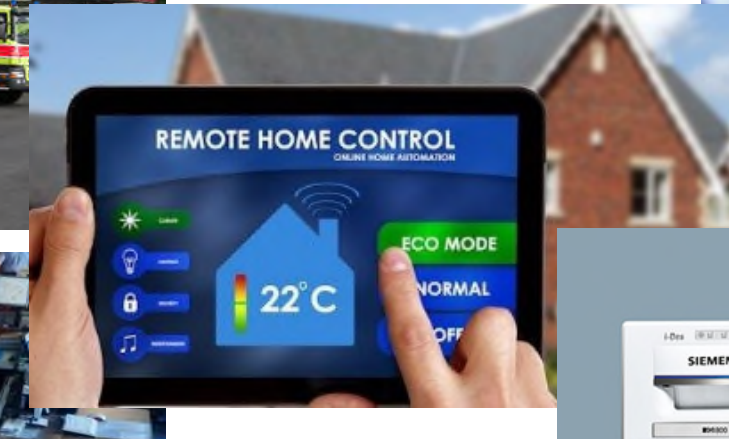
<http://kardiologie-potsdam.de>



<http://techpp.com>

GPS-Navigationsgerät

... und noch viel mehr



<http://codeverge.com>



... unsere Abhängigkeit von Software ist nahezu total geworden !

Software

fehlerfrei

fehlerhaft

Erfolgsgeschichte

Katastrophe

<http://www.summitpost.org/the-grat/559212>



Software-**Erfolgsgeschichten:**
6 Beispiele

Software Erfolgsgeschichte: Beispiel 1 Mercedes GLE 500e (Hybrid)



Software-gesteuerte Funktionen:

- Hybridoptimierung (Elektro/BenzinV6)
- Elektronisches Stabilitätsprogramm
- Verschiedene Assistenten (Spurhalten, Totwinkel, Abstand, ...)
- ...

Software Erfolgsgeschichte: Beispiel **1**
Mercedes S500



Autonome Langstreckenfahren

August 2013: Das “S 500 INTELLIGENT DRIVE”
Versuchsfahrzeug fährt autonom 100 Kilometer von
Mannheim nach Pforzheim

Software Erfolgsgeschichte: Beispiel 1
2016: Erstes Rennen mit *fahrerlosen* E-Cars



<http://www.telegraph.co.uk>

Teilnehmer: 20 mechanisch *identische* E-Cars mit mehr als 300 km/h Spitzengeschwindigkeit

Entscheidend: *Software* (Algorithmen, künstliche Intelligenz, Lernfähigkeit, ...)

Software Erfolgsgeschichte: Beispiel 2 Künstliche Intelligenz spielt «GO»

<https://fr.wikipedia.org>



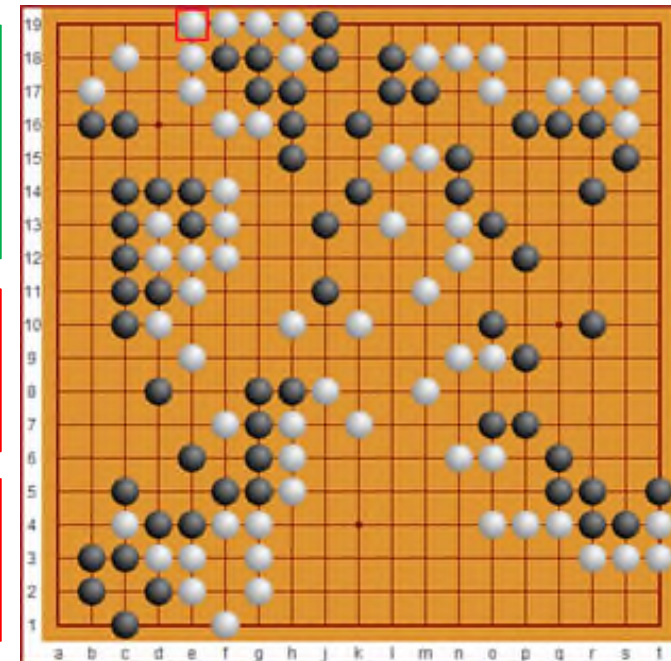
«GO» ist ein Strategie-Brettspiel, das vor ca. 2`500 Jahren in China erfunden wurde.

Spielbrett: 19 x 19 Linien, unbeschränkte Anzahl schwarzer und weisser Steine.

Ziel: Möglichst grosse Gebietsanteile besetzen

Anzahl verschiedener Stellungen auf dem GO-Brett: $\sim 4,63 \times 10^{170}$

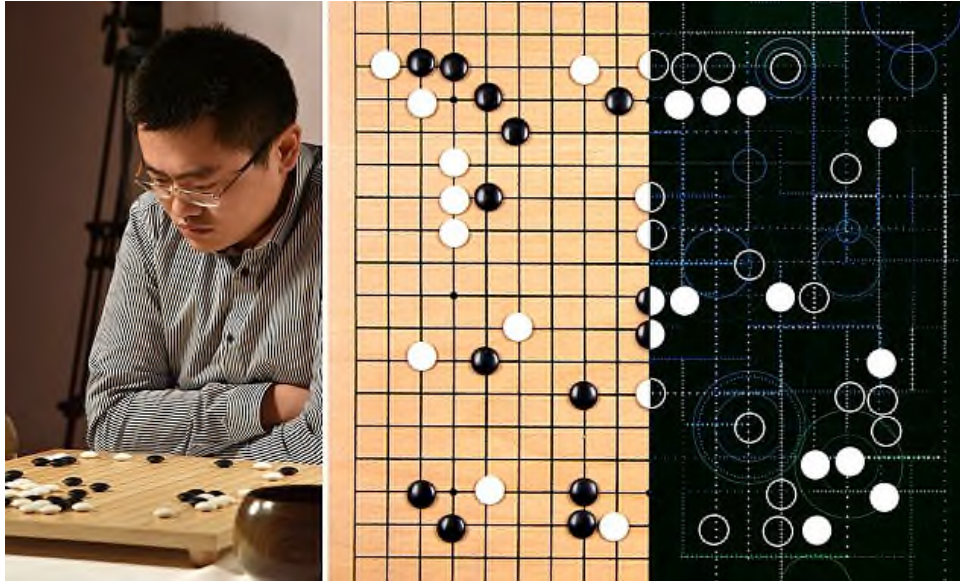
- Schach: $\sim 10^{43}$
- Anzahl Atome im Universum: $\sim 10^{80}$



<http://www.brettspielnetz.de>

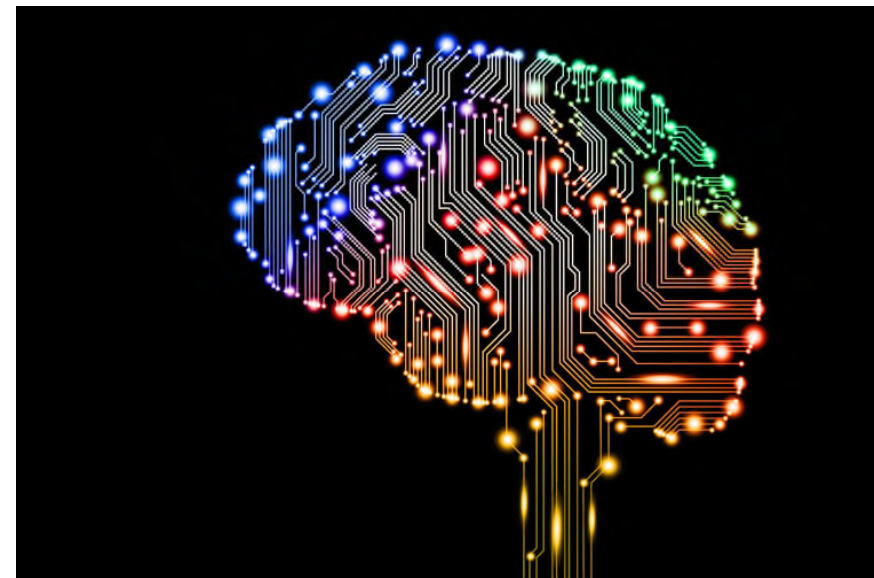
Software Erfolgsgeschichte: Beispiel 2 Künstliche Intelligenz spielt «GO»

<http://www.dailymail.co.uk>



Januar 2016: Das KI-Programm «AlphaGO» gewinnt gegen den mehrfachen und amtierenden GO-Europameister Fan Hui 5:0

Beeindruckend/Erschreckend:
«AlphaGO» wurde NICHT programmiert, sondern ist ein *selbstlernendes* Programm
[Deep Learning]



<http://www.digitaltrends.com>

Software Erfolgsgeschichte: Beispiel **2**
Künstliche Intelligenz spielt «GO»

Beeindruckend/Erschreckend:

«AlphaGO» wurde NICHT programmiert, sondern ist ein *selbstlernendes* Programm [Deep Learning]

<http://www.andreaskundert.ch>



**Da wurde eine Hürde
übersprungen, welche
unvorhersehbare
Konsequenzen haben
wird**

Software Erfolgsgeschichte: Beispiel 3 Krebsdiagnose/Therapie-System IBM Watson

<http://de.slideshare.net>

The screenshot displays the IBM Watson Oncology Advisor interface. It features a table with three columns: Treatment Plan, Confidence, and Patient Preferences Match. The first plan has a 95% confidence score and is labeled 'Acceptable'. The second plan has a 45% confidence score and is labeled 'Unacceptable'. The third plan has an 8% confidence score and is labeled 'Preferred'. Below the table, a note states 'Radiation and Surgery are unlikely to be appropriate.' The IBM logo is in the top left, and the WATSON logo is in the top right. The text 'IBM Watson Oncology Advisor' is prominently displayed in the center-left. At the bottom, there is a disclaimer and the year 2012.

Treatment Plan	Confidence	Patient Preferences Match
Treatment plan 1 Roxatone, Oxycotin, Cyclophosphamide, Docetaxel	95%	Acceptable match with patient preferences
Treatment plan 2 Systemic Chemotherapy, Herceptin, Docetaxel	45%	Unacceptable match with patient preferences
Treatment plan 3 Systemic Chemotherapy	8%	Preferred match with patient preferences

IBM Confidential: References to potential future products are subject to the Important Disclaimer provided earlier in the presentation

14 © 2012 IBM Corporation

Software-Expertensystem für Krebsdiagnose und Krebstherapie
Kennt und «versteht» das gesamte onkologische Wissen

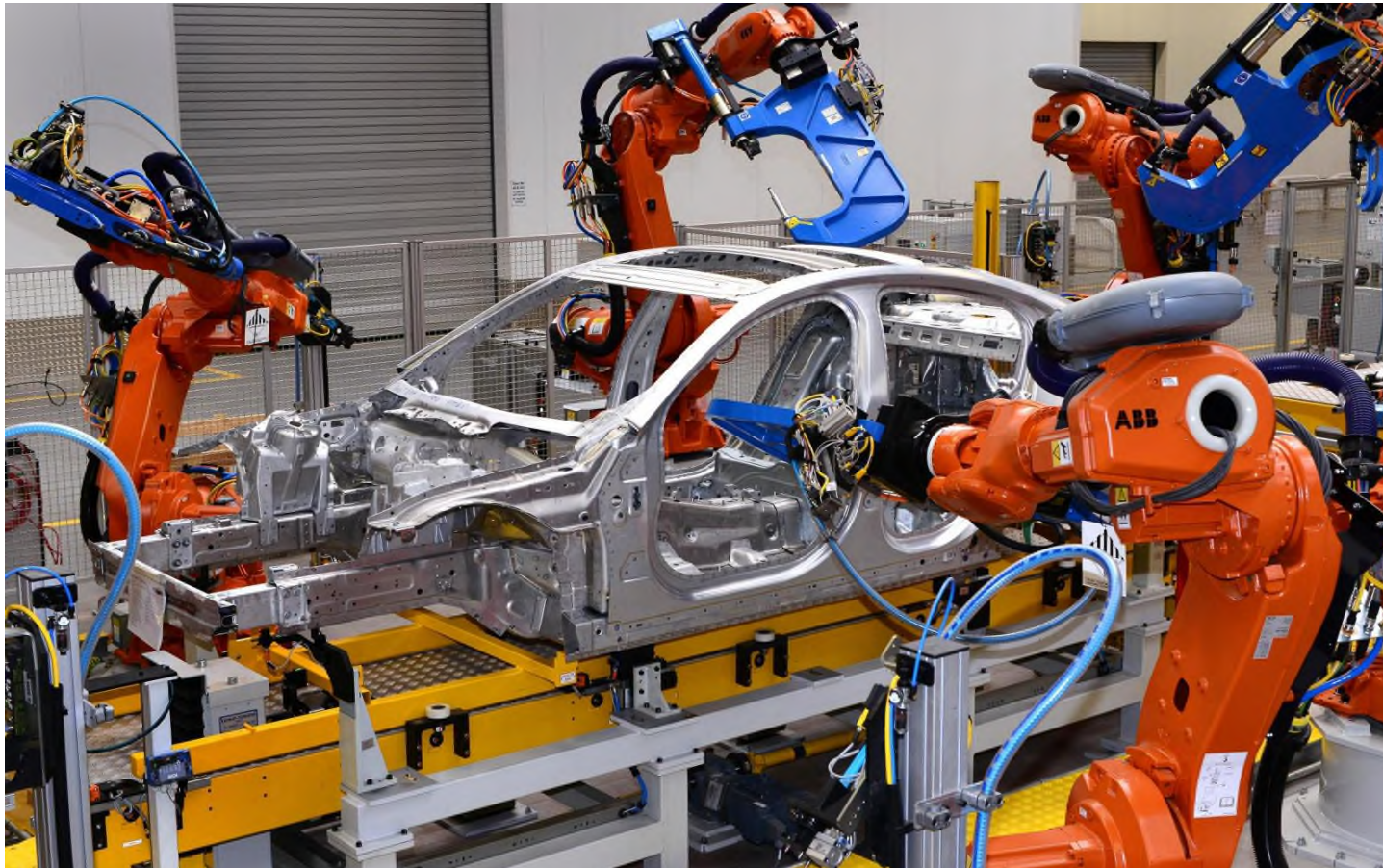
Software Erfolgsgeschichte: Beispiel **3** Krebsdiagnose/Therapie-System IBM Watson



Software-gesteuerte Funktionen:

- Enorme Wissensbasis
- Interaktive Diagnose-/Therapieführung
 - Personalisierte Medizin
 - ...

Software Erfolgsgeschichte: Beispiel 4 Vollautomatisierte Fahrzeugproduktion



www.metalworkingworldimagazine.com

Schweiss-/Montageroboter

Software Erfolgsgeschichte: Beispiel **4**
Vollautomatisierte Fahrzeugproduktion



<http://we.bofthings.org>

Menschenleeres Produktionsfließband
«Dunkle Fabrik»

Software Erfolgsgeschichte: Beispiel 5 e-Commerce



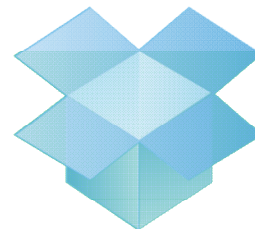
Software-gesteuerte Funktionen:

- Produktkatalog/Personalisierte Empfehlungen
 - Warenkorb/Zahlungsabwicklung
 - Logistik
 - ...

Software Erfolgsgeschichte: Beispiel 5
e-Commerce

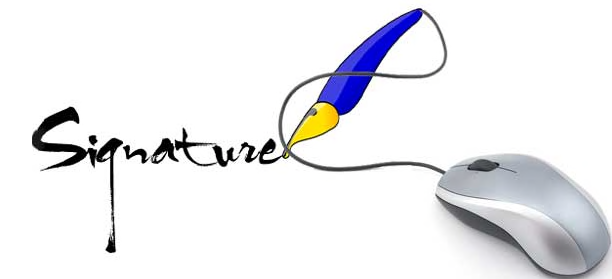


Electronic Social Networks



Dropbox

Cloud Document
Storage



Software Erfolgsgeschichte: Beispiel 6 Pflegeroboter

<http://www.blick.ch>



Roboter für die Pflege und
Altenbetreuung erreichen
die Einsatzbereitschaft



<http://www.wiwo.de>

Software Erfolgsgeschichte: Beispiel **6**
Pflegeroboter



<http://www.smh.com.au>

... sie werden zu unseren täglichen, geschätzten Begleitern werden



Software-**Katastrophen**:
6 Beispiele

Software Katastrophe: Beispiel **1**
Absturz Airbus A400M (9. Mai 2015)

<http://www.reuters.com>



A400M: Militärisches Transportflugzeug
Zuladung von 37 Tonnen, Reichweite über 3'000 km

Software Katastrophe: Beispiel **1**
Absturz Airbus A400M (9. Mai 2015)

<http://www.ouest-france.fr>



Ausfall der
Schubsteuerung von 3
Triebwerken nach dem
Start ⇒ **Absturz**

Grund: Unvollständige Triebwerkdaten nach Softwareupdate
⇒ Software darf NIE ohne Datenprüfung aufstarten

Software Katastrophe: Beispiel 2

US Home Depot Kreditkartendiebstahl (September 2014)



<http://www.dailytech.com>



<http://www.aprinder.com>

“Mehr als 40 Millionen Kundenkreditkarten von russischen Hackern gestohlen”
(⇒ später auf dem Schwarzmarkt angeboten)

Software Katastrophe: Beispiel **3**
Knight Capital 440 M\$ Handelsverlust (1. August 2012)

<http://bilder1.n-tv.de>



Knight Capital:
Computer-Trader
im Hochfrequenzhandel

[10'000 Trades/sec
Haltezeit: Millisekunden]

Handelsverlust am 1.8.2012 (NYSE): **440 Millionen US\$**

Software Katastrophe: Beispiel 3

Knight Capital 440 M\$ Handelsverlust (1. August 2012)

<http://www.nj.com>



Am 1.8.2012 um 9:30 generierten die Computer (ohne menschliches Zutun) Millionen von fehlerhaften Trades

Um 9:58 hatte Knight Capital 440 Millionen US\$ verloren

Grund: Programmierfehler im automatisierten High-Frequency Trading-Algorithmus der Firma (nach einem Software-Update)

Software Katastrophe: Beispiel 4
Einbruch in Bundestag-Intranet (Mai 2015)



<http://www.faz.net/>

Die Bundestags-Computer sind im Mai 2015 Ziel einer bislang beispiellosen Attacke noch unbekannter Hacker geworden. Wie stark und mit welchen Folgen das Datennetz des Parlaments in Mitleidenschaft gezogen ist, bleibt zunächst unklar.

Software Katastrophe: Beispiel 4 Einbruch in Bundestag-Intranet (Mai 2015)

<http://blog.softwaredia.com>



Es wurden grosse Mengen
vertraulicher E-Mails von
Abgeordneten erbeutet

Bisherige Erkenntnisse: Daten von
rund 16 Gigabyte kopiert



Software Katastrophe: Beispiel 5

Remote Car Hacking (Jeep Model 2014, July 2015)



<https://www.dotcomsecurity.de/2015/07/>

Charlie Miller und Chris Valasek gelang es 2015 einen serienmässigen Jeep (Modelljahrgang 2014) von ihrem Wohnzimmer aus zu hacken und die Kontrolle über dessen vitale Funktionen zu übernehmen

Software Katastrophe: Beispiel **5**
Remote Car Hacking (Jeep Model 2014, July 2015)



<http://craigpeterson.com/automotive/hacking>

Unmittelbares Resultat: Chrysler Rückruf (1.4 Mio Fz),
Sammelklagen und hektische Aktivität im US-Kongress

Software Katastrophe: Beispiel 6 Ungewollte Beschleunigung bei Toyota Fahrzeugen



<http://businessethiccases.blogspot.ch>

Die ungewollte Beschleunigung von Toyota und Lexus Fahrzeugen führte von 2000-2010 zu **89 Verkehrstoten** und **52 Verletzten**

Software Katastrophe: Beispiel 6 Ungewollte Beschleunigung bei Toyota Fahrzeugen



<http://www.autoevolution.com>

Toyota behauptete anfänglich, Grund für die ungewollte Beschleunigung sei die **Fussmatte**

Unabhängige Untersuchungen wiesen aber ein **Softwareproblem** in der Steuerung des Gaspedales nach

19. März 2014: Toyota bezahlt eine Busse von 1.2 Milliarden US\$



Unsere Abhängigkeit von Software ist nahezu total geworden !

Müssen wir mit den Software-Risiken leben?

Was können wir dagegen unternehmen?


```
1
2 public class PassByValue{
3     public static class Holder{
4         public String value;
5     }
6
7     public static void passByValue( Holder holder){
8         holder.value = "changed";
9     }
10
11    public static void main( String[] args ){
12        Holder holder = new Holder();
13        holder.value = "test";
14        passByValue( holder );
15        System.out.println( holder.value );
16    }
17 }
18 }
```

Widerstandsfähige
(resiliente)
Software

Fehlerbehaftete
Software

Erfolgs-
geschichte

Katastrophe

Wie entsteht Software?

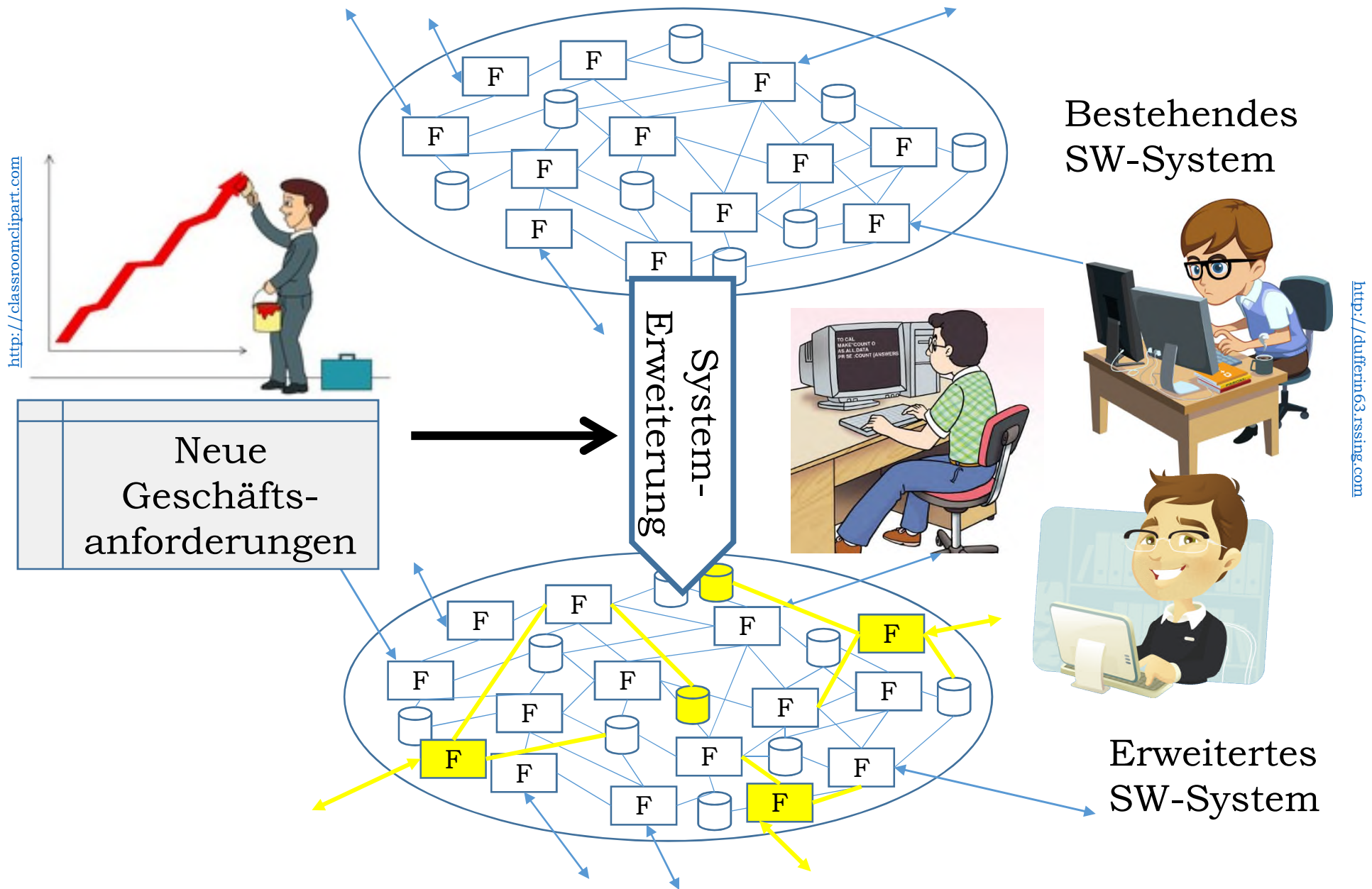
... noch eine kurze Einführung für Nicht-Informatiker





«Software Development is a technical activity
conducted by human beings»

Niklaus Wirth [ETHZ]



<http://www.business-netz.com>



System-
Erweiterung

Widerstandsfähige
(resiliente)
Software

Software-Entwicklungsprozess

```
1 public class PassByValue {
2     public static class Holder {
3         public String value;
4     }
5 }
6
7 public static void passByValue( Holder holder){
8     holder.value = "changed";
9 }
10
11 public static void main( String[] args ){
12     Holder holder = new Holder();
13     holder.value = "test";
14     passByValue( holder );
15     System.out.println( holder.value );
16 }
17 }
18
```

⇒ Fehler & Fehlentscheide in allen Phasen
des Software-Entwicklungsprozesses

Fehlerbehaftete
Software

Wie bauen wir widerstandsfähige (resiliente) Software?

Wie vermeiden wir fehlerbehaftete Software?

Widerstandsfähige (resiliente) Software

Software-Entwicklungsprozess

Antwort:

- Adäquate **Strategie**
- Korrekte Prinzipien & Prozesse
- Geeignete Informatiker
- Qualitäts-Firmenkultur



```
1 public class PassByValue {
2     public static class Holder {
3         public String value;
4     }
5
6     public static void passByValue( Holder holder){
7         holder.value = "changed";
8     }
9
10
11
12     public static void main( String[] args ){
13         Holder holder = new Holder();
14         holder.value = "test";
15         passByValue( holder );
16         System.out.println( holder.value );
17     }
18 }
```

Fehlerbehaftete Software



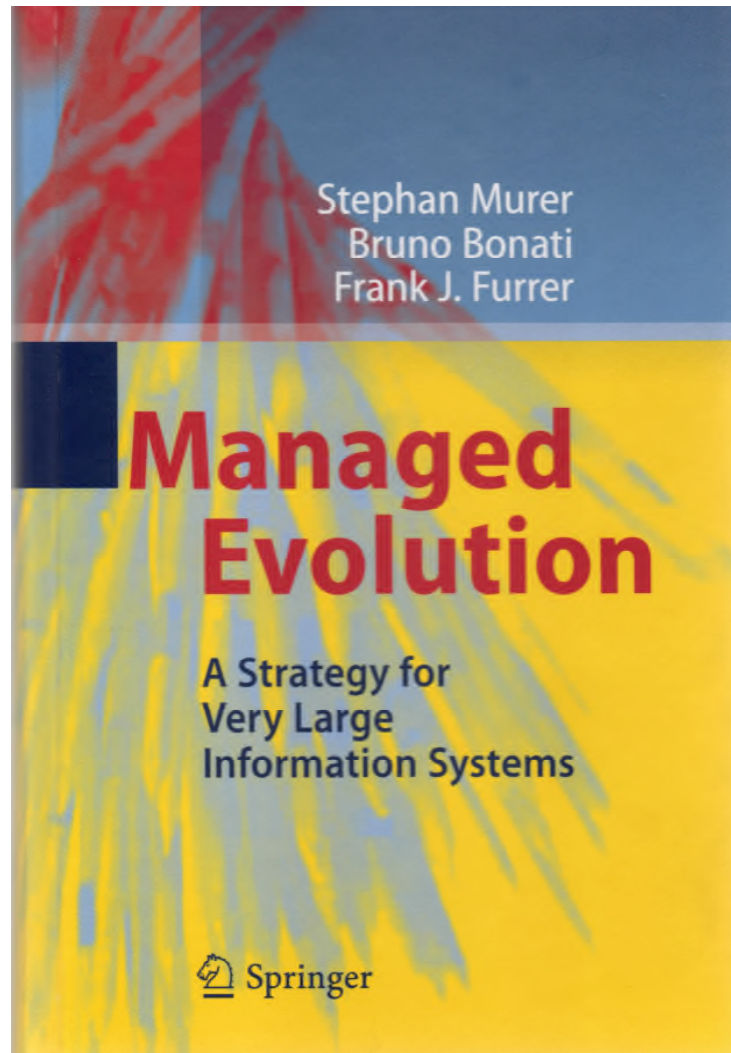
Software-Entwicklungsstrategie:

MANAGED EVOLUTION

... jetzt wird es etwas akademischer

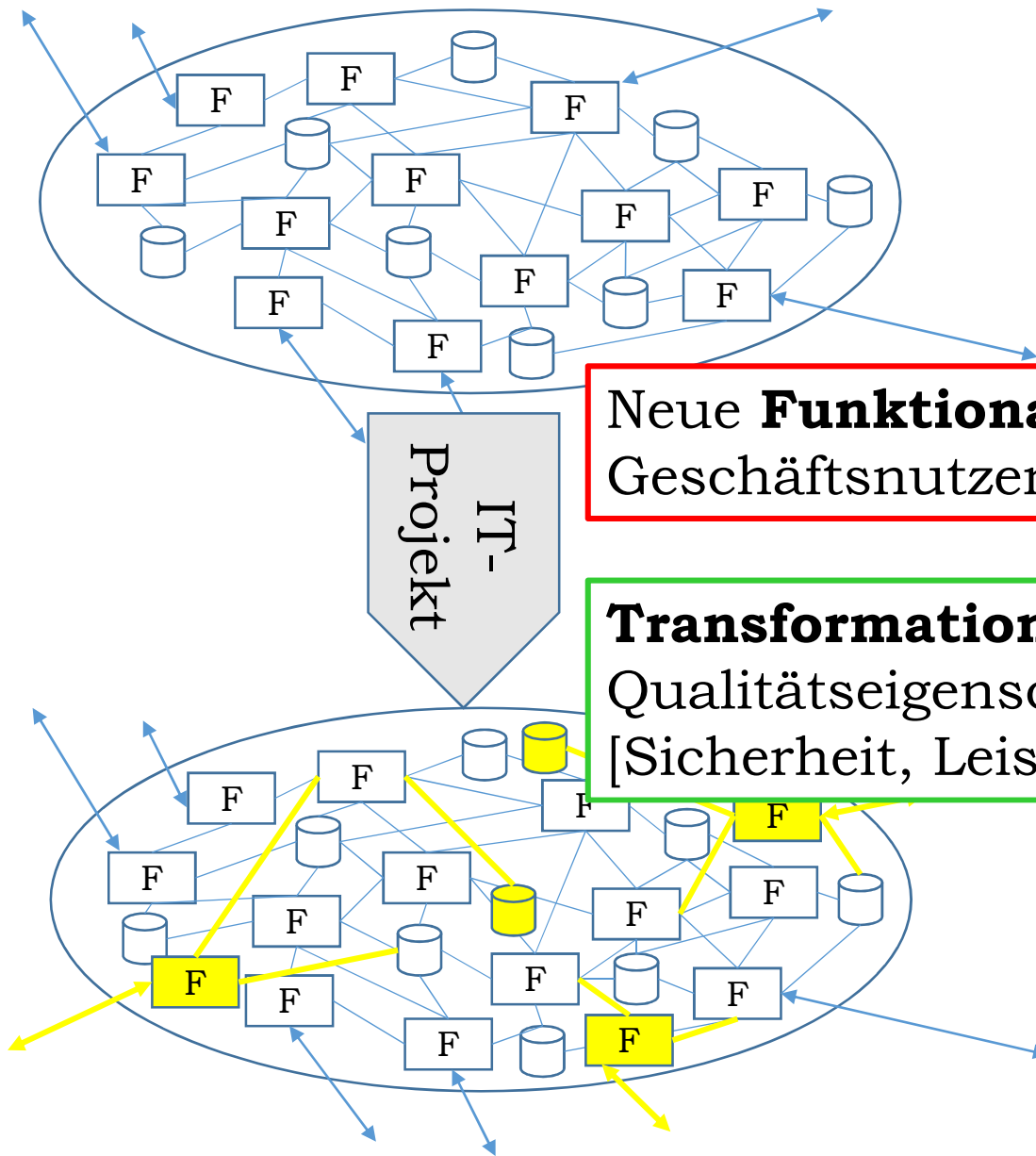
MANAGED EVOLUTION

ISBN 978-3-642-01632-5, 2011



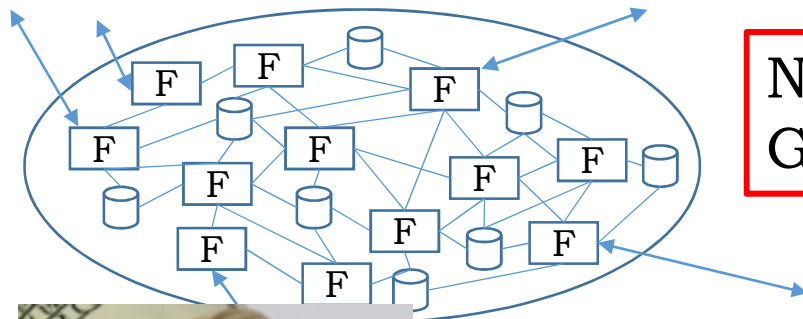
= Software Entwicklungsstrategie

- Laufende Erhöhung des Geschäftsnutzens
- Kontinuierliche Verbesserung der **Agilität** und der **Resilience**
- Vertrauensvolle Zusammenarbeit zwischen den Geschäftseinheiten und den IT-Abteilungen



Neue **Funktionalität** = Erhöhung des Geschäftsnutzens (Business Value)

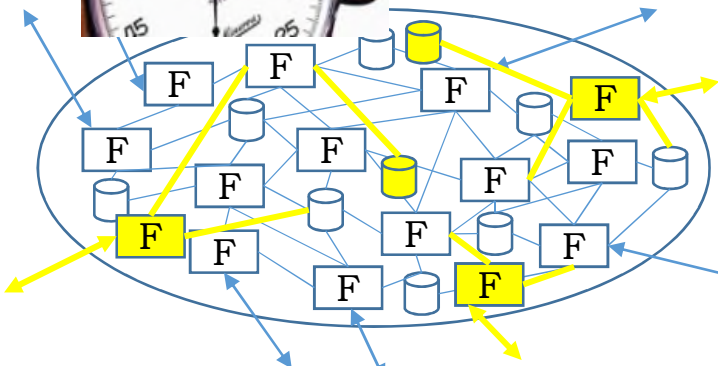
Transformation der Qualitätseigenschaften [Sicherheit, Leistungsdaten, ...]



Neue **Funktionalität** = Erhöhung des Geschäftsnutzens (Business Value)



IT-Projekt



<http://www.clipartsheep.com>

Transformation der Qualitätseigenschaften [Sicherheit, Leistungsdaten, ...]



Transformation der **Qualitätseigenschaften**
[Sicherheit, Zuverlässigkeit, Integrität, Leistungsdaten, ...]



Verbesserung



Verschlechterung

Wir haben bei jedem
IT-Projekt die Wahl

IT-Projekt



Transformation der **Qualitätseigenschaften**
[Sicherheit, Zuverlässigkeit, Integrität, Leistungsdaten, ...]



Verschlechterung





Managed Evolution

[Murer, Bonati, Furrer - 2011]

Geschäftsnutzen [Business Value]

Agility [Agilität]

Widerstandsfähigkeit [Resilience]

Credo:

Müssen kontinuierlich
- mit jedem Projekt –
verbessert werden

... alle anderen Qualitätseigenschaften

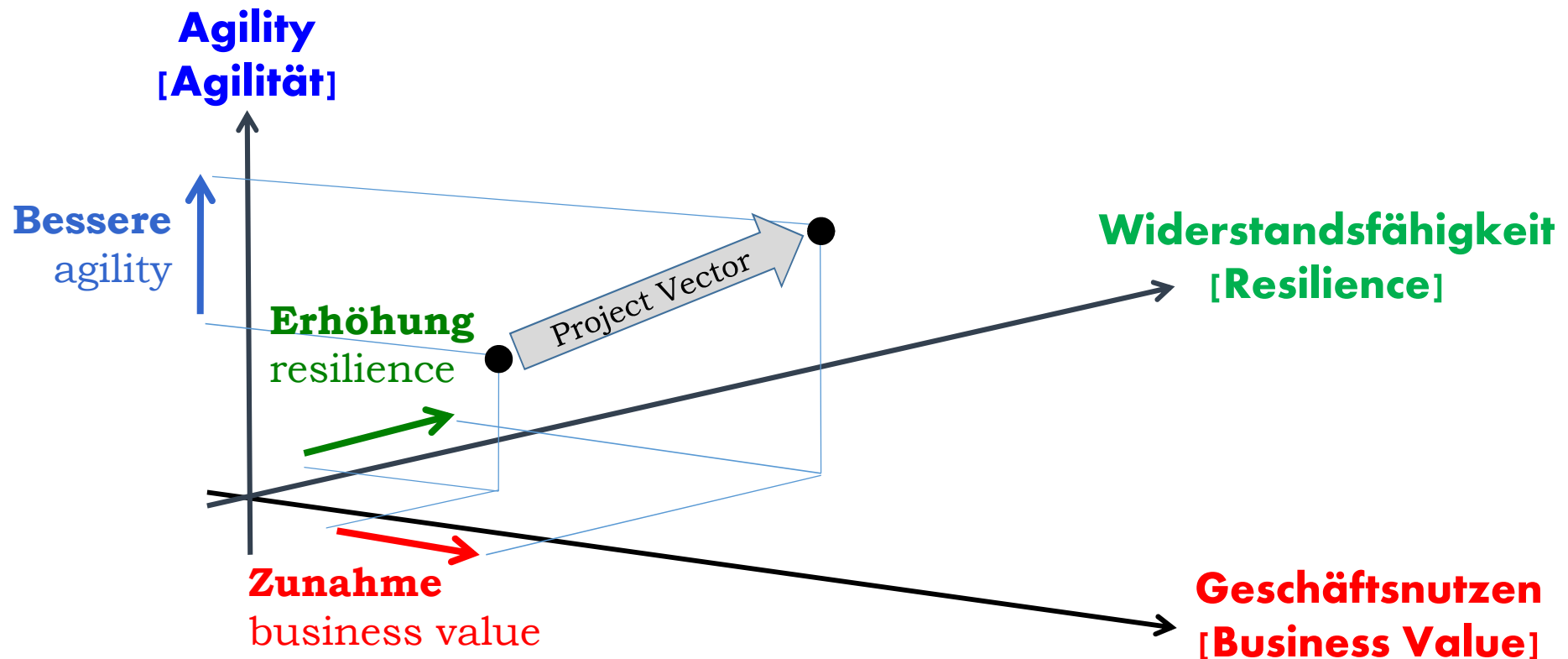
So gut wie nötig



Managed Evolution

[Murer, Bonati, Furrer - 2011]

Managed Evolution: 3 messbare Schlüsselqualitäten des SW-Systemes

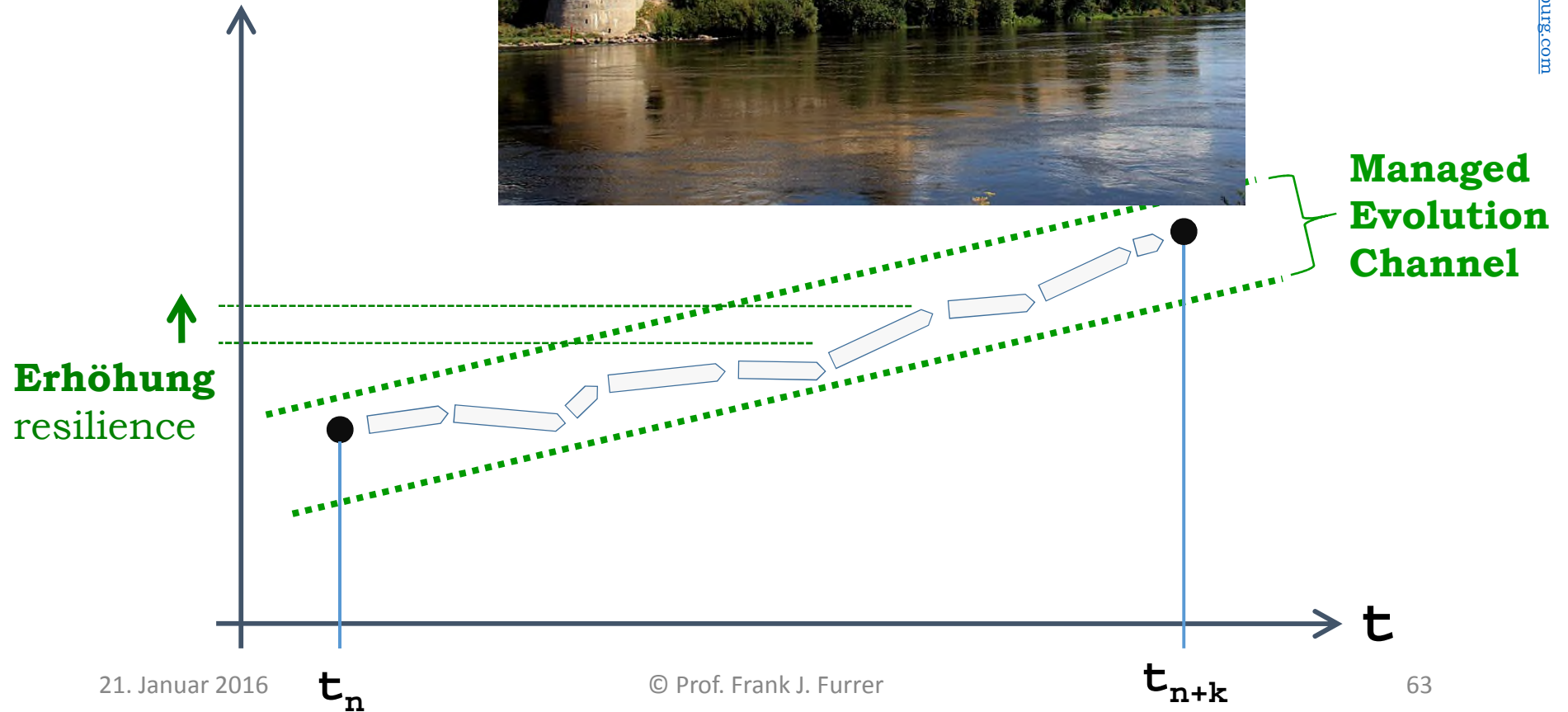


Resilience = Widerstandsfähigkeit gegen Fehler, Ausfälle, Angriffe, ...



<http://www.saint-petersburg.com>

**Widerstandsfähigkeit
[Resilience]**





... einleuchtend – aber gibt es Widerstände?

- Verständnis
- Vertrauen

Geschäftseinheiten wollen:

- Kurze Markteinführungszeit
- Tiefe Entwicklungskosten
- Nur die essentielle Funktionalität
- Neueste Technologien

CIO & IT-Architektur wollen:

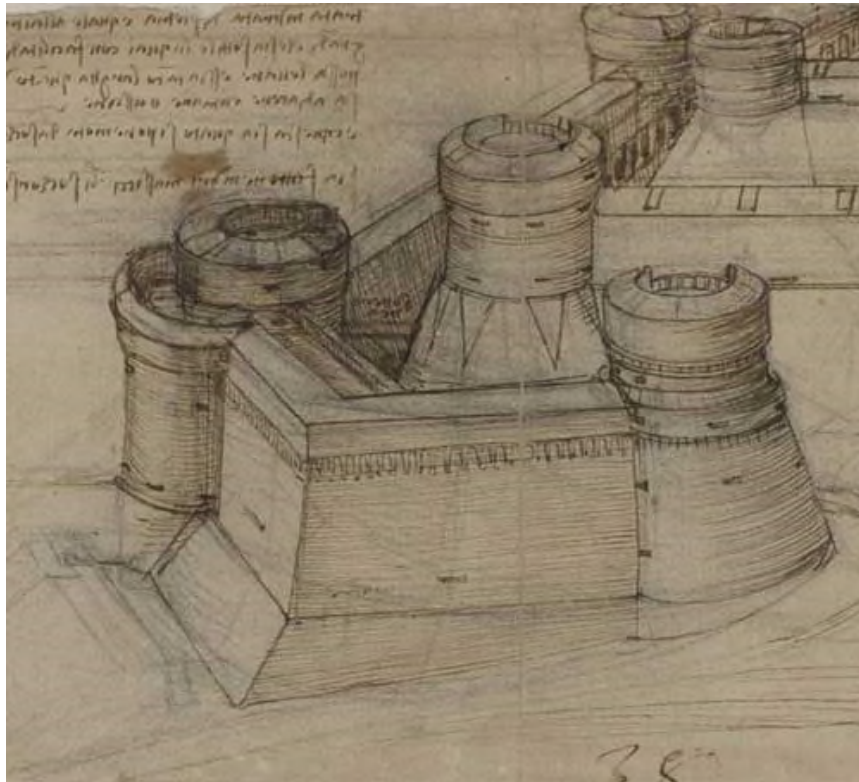
- Verbesserung der Agility
- Erhöhung der Resilience
- Beschränkung der Komplexität
- Keine technischen Schulden





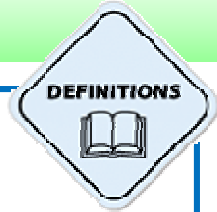
Kernfrage: **Wie** baut man resiliente Systeme?

<http://www.marinabaysands.com>

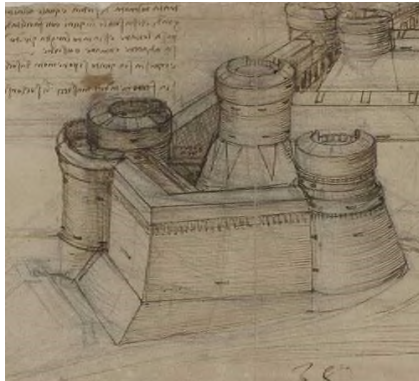


Fundament =
Architektur

[System- & Softwarearchitektur]



<http://www.marinabaysands.com>



Architektur =

Struktur und Beziehungen der Elemente in einem System, sowie die Beziehungen zur Umwelt

<http://www.skyscrapercity.com>



Resilienz:

- Verkehrsstau
- Stromausfälle
- Hochwasserschutz
- Feuerwehrzugang
- Polizeistandorte
- Krankenhausdichte
- ...

Gibt es eine Medizin für resiliente Architektur?

<http://www.laurelleaffarm.com>



Architektur-Prinzipien

Fundamentale Einsichten – als Regeln formuliert – wie ein gutes Softwaresystem gebaut wird («Ewige Wahrheiten»)

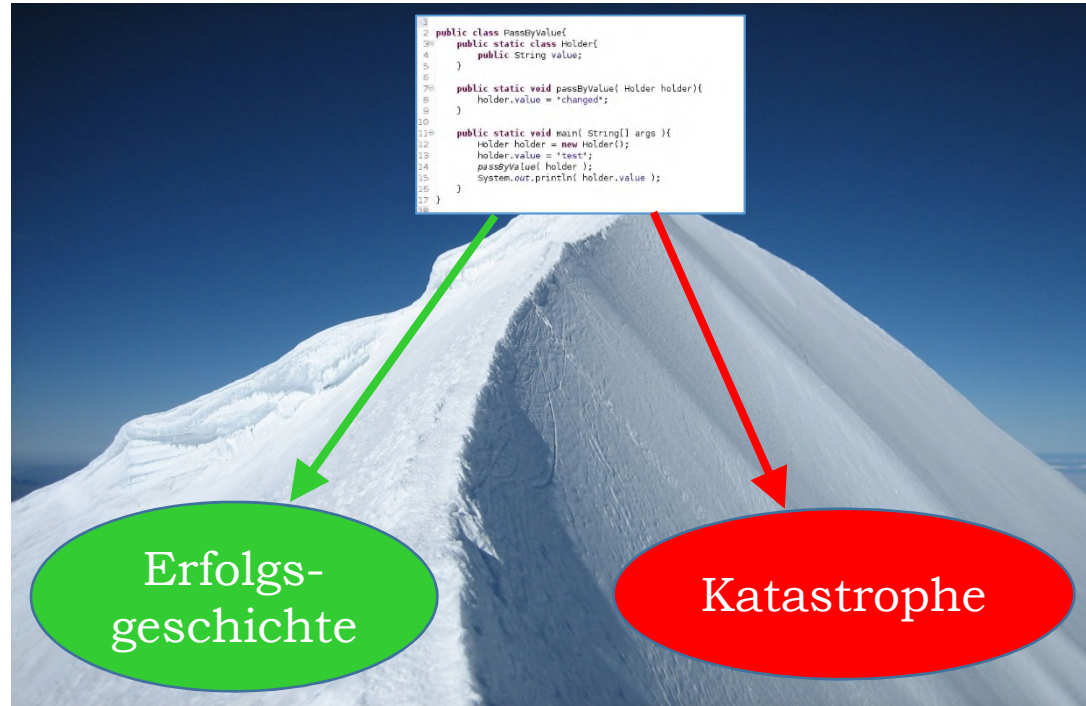


Die *strikte* und *konsequente* Anwendung der Architektur-Prinzipien führt zu agilen und resilienten Systemen

<http://wordsmithmaverickartista.com/>



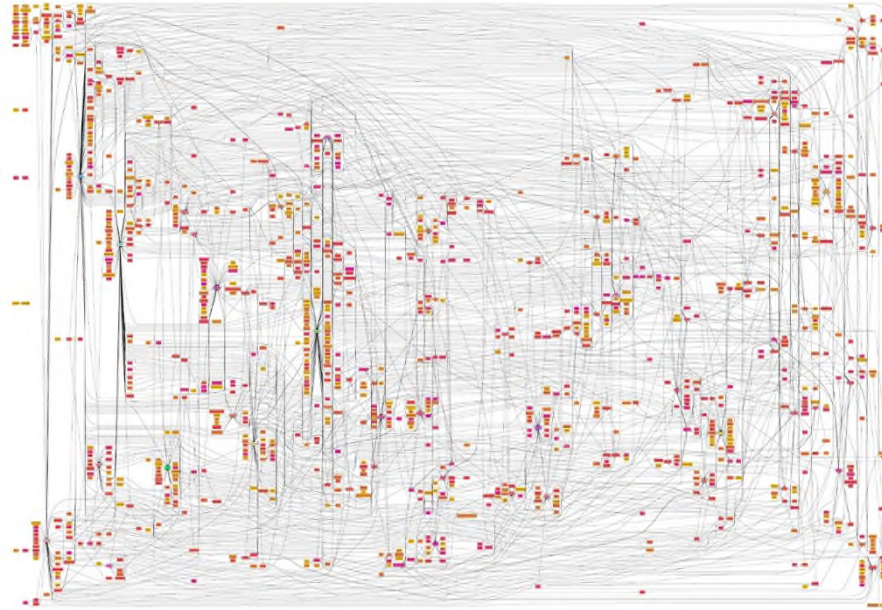
Schlussfolgerungen



Müssen wir mit den Software-Risiken leben?

Was können wir dagegen unternehmen?

Müssen wir mit den **Software-Risiken** leben?



Die heutigen **Software-Systeme** sind die komplexesten technischen Systeme in der Welt



... daher müssen wir ein **Restrisiko** akzeptieren

Was können wir gegen die Risiken unternehmen?

müssen

- ✓ Strikte Einhaltung der Architekturprinzipien
- ✓ Verlässliches SW-Entwicklungsvorgehen (Managed Evolution)
- ✓ Grundlegende Sicherheitskultur in Software-Firmen
- ✓ Wirksamere Produkt-Haftungsgesetze für Software (weltweit)
- ✓ ... und eine exzellente Ausbildung unserer Informatiker



<http://www.ebay.com>



Ausgewählte Literaturangaben



Literatur [1]

TU Dresden	Die Technische Universität Dresden http://tu-dresden.de
Software-Grundlagen	Jochen Ludewig, Horst Lichter: Software Engineering - Grundlagen, Menschen, Prozesse, Techniken Dpunkt verlag GmbH, 3. Auflage, 2013. ISBN 978-3-8649-0092-1
Mercedes-Benz autonomous driving	Mercedes-Benz long-distance autonomous driving https://www.mercedes-benz.com/en/mercedes-benz/innovation/autonomous-long-distance-drive/
Computer GO	Sarah Griffiths: Artificial intelligence breakthrough as Google's software beats grandmaster of Go, the 'most complex game ever devised' Daily Mail Online, January 31, 2016 http://www.dailymail.co.uk/sciencetech/article-3419548/March-machines-Computer-BEATS-one-world-s-best-players-ancient-board-game-using-human-like-skills.html
IBM Watson Cancer Expert System	IBM Watson for Oncology - Get oncologists the assistance they need to make more informed treatment decisions http://www.ibm.com/smarterplanet/us/en/ibmwatson/watson-oncology.html und: John E. Kelly, Steve Hamm: Smart Machines – IBM's Watson and the Era of Cognitive Computing Columbia University Press, New York, N.Y., USA, 2013. ISBN 978-0-231-16856-4
Pflegeroboter	Stefan Simons: Pflegeroboter - Hightech-Kameraden fürs Alter SPIEGEL On-Line, Freitag, 22.03.2013 – 18:40 Uhr http://www.spiegel.de/gesundheit/diagnose/pflegeroboter-hightech-kameraden-fuers-alter-a-890342.html



Literatur [2]

Knight Capital	Edgar Perez: Knightmare on Wall Street – <i>The Rise and Fall of Knight Capital and the biggest Risk for Financial Markets</i> www.mredgarperez.com , 2013. ISBN 978-0-9896577-0-9
Bundestag-Hacking	Maik Baumgärtner, Sven Röbel, Wolf Wiedmann-Schmidt: Cyberangriff auf den Bundestag - <i>Hacker kopierten Abgeordneten-E-Mails</i> SPIEGEL On-Line, Donnerstag, 18.06.2015 http://www.spiegel.de/politik/deutschland/cyberangriff-auf-bundestag-abgeordneten-e-mails-erbeutet-a-1039388.html
Toyota Case	Junko Yoshida: Acceleration Case - <i>Jury Finds Toyota Liable</i> EE Times, 10/24/2013 09:00 PM EDT http://www.eetimes.com/document.asp?doc_id=1319897 und auch: https://en.wikipedia.org/wiki/Sudden_unintended_acceleration
Jeep Hacking	Charlie Miller, Chris Valasek: <i>Remote Exploitation of an Unaltered Passenger Vehicle</i> . August 10, 2015. Downloadable from: http://illmatics.com/Remote%20Car%20Hacking.pdf [last accessed: 17.8.2015]



Literatur [3]

Vorlesung Prof. Furrer, WS15/16	Prof. Dr. Frank J. Furrer: Future-Proof Software Systems Lecture Winter Term 2015/16 http://tu-dresden.de/die_tu_dresden/fakultaeten/fakultaet_informatik/smt/st/studium?leaf=1&lang=en&subject=304
Managed Evolution	Stephan Murer, Bruno Bonati, Frank J. Furrer: Managed Evolution – A Strategy for Very Large Information Systems Springer-Verlag, Berlin Heidelberg, 2011, ISBN 978-3-642-01632-5
Langlebige Software	Carola Lilienthal: Langlebige Softwarearchitekturen – Technische Schulden analysieren, begrenzen und abbauen Dpunkt Verlag, Heidelberg, 2016. ISBN 978-3-86490-292-5
Zukunftsfähige Software	Frank J. Furrer: Zukunftsfähige Softwaresysteme – Zukunftsfähig trotz zunehmender SW-Abhängigkeit Informatik Spektrum, Springer-Verlag, Heidelberg. First online: 30 June 2015. 10.1007/s00287-015-0909-6. Downloadable from SpringerLink: http://link.springer.com/article/10.1007/s00287-015-0909-6 (Last accessed: 31.12.2015)
Architecture	George Fairbanks: Just Enough Software Architecture – A Risk-Driven Approach Marshall & Brainerd, Boulder CO, USA, 2010. ISBN 978-0-9846181-0-1
Secure Architecture	Eduardo B. Fernandez: Security Patterns in Practice – Designing Secure Architectures using Software Patterns John Wiley & Sons, Ltd., Chichester UK, 2013. ISBN 978-1-119-99894-5
O. N. Bradley	Omar Nelson Bradley (* 12. Februar 1893 in Clark, Missouri, † 8. April 1981 in New York City) https://de.wikipedia.org/wiki/Omar_N._Bradley http://www.brainyquote.com/quotes/authors/o/omar_n_bradley.html



TECHNISCHE
UNIVERSITÄT
DRESDEN

Der Rektor

Hiermit bestelle ich

HERRN

DR. SC. TECHN. FRANK J. FURRER

gemäß § 65 Abs. 2
Sächsisches Hochschulfreiheitsgesetz (SächsHSFG)
i.d.F. vom 15. Januar 2013

mit Wirkung vom 01. Juli 2015

zum

HONORARPROFESSOR

für

SOFTWAREARCHITEKTUR GROSSER SYSTEME

Gemäß § 65 Abs. 3 SächsHSFG sind Honorarprofessoren berechtigt,
für die Dauer ihrer Bestellung den akademischen Titel
„Professor“ zu führen.

Dresden, den 23. Juni 2015



Prof. Dr.-Ing. habil. DEng/Auckland
Hans Müller-Steinhagen

Mein **herzlicher Dank** geht
an die Fakultät für
Informatik der Technischen
Universität Dresden

Es ist für mich eine
wunderbare Erfüllung, meine
«letzte Runde» im beruflichen
Leben mit interessanten
Kollegen und motivierten
Studenten verbringen zu
dürfen



If we continue to develop our technology without wisdom or prudence, our servant may prove to be our executioner

Omar N. Bradley (U.S. Army General, Chairman of the Joint Chiefs of Staff [1949])

Danke für Ihre Aufmerksamkeit



Fragen ?