

Willkommen zur Vorlesung  
*Softwarekonstruktion*  
im Wintersemester 2011 / 2012  
Prof. Dr. Jan Jürjens

TU Dortmund, Fakultät Informatik, Lehrstuhl XIV

Vorlesungswebseite (bitte notieren):  
<http://www-jj.cs.tu-dortmund.de/jj/teaching/ws11/swk>



- Organisatorisches
- Vorstellung des Fachgebietes
- Vorlesungsinhalte

- Studienordnung
  - Einordnung / Kompetenzen / Struktur / Prüfung
- Vorlesung
  - Bildungsvertrag, Termine, Feedback
- Übung
  - Konzept / Termine
- Klausur

- Bachelor Informatik / Angewandte Informatik
  - Wahlpflichtmodul
  - Teilnahmevoraussetzungen
    - Erfolgreich abgeschlossenes Modul „Software-Technik“ (SWT)
    - Kenntnisse: Objektorientierung, Programmierpraxis

- Erlangbare Kompetenzen innerhalb der Vorlesung:
  - grundlegende Prinzipien der ingenieurmäßigen Konstruktion von Software
  - ihre Vor- und Nachteile für eine gegebene Problemstellung und deren Anwendung auf Probleme mittlerer Größe
  - formale Spezifikationssprachen zur Beschreibung der Architektur von Systemen
  - Ansätze aus der Logik zur Überprüfung von Systementwürfen
  - die Organisation großer Softwaresysteme unter Prüfung verschiedener Gesichtspunkte

- 3 SWS:
  - 2 SWS Vorlesung
  - 1 SWS Übung
- 4 Credits
  - 3 Credits Vorlesung
  - 1 Credits Übung
- **Aufwand 120 Stunden**
  - 45 Stunden Präsenz ( $15 \cdot (2+1)$ )
  - 75 Stunden Vor-/Nachbereitung und Hausübungen ( $15 \cdot 5$ )
- Veranstaltungssprache Deutsch

- Diplom
  - Prüfungsleistung: Klausur
  
- Bachelor
  - Studienleistung: Übungsschein
  - Modulprüfung: Klausur (bei bestandener Studienleistung)

- Fachliche Einführung in das Thema Softwarekonstruktion
- Engagierte Betreuung
  - Interessante Vorlesung
  - Regelmäßige Sprechstunden (Termin s. Homepage)
  - Betreute Übungen
  - Korrigierte Hausübungen
  - Transparente Anforderungen
  - Möglichkeiten zum direkten Feedback
- Möglichkeit zum Erwerb des Scheins



- Aktives Auseinandersetzen mit den Vorlesungsinhalten
  - Aktive Teilnahme an der Vorlesung
  - Vor- und Nachbereitung der Vorlesung
  - Aktive Teilnahme an den Übungen
  - Bearbeitung der Hausübungen

- Termine:
  - Mo. 16:00 bis 18:00 Otto-Hahn-Str. 14 – E23  
5-min. Pause ca. um 17:00
- **Zu beachten:**
  - Um eventuelle Terminausfälle zu kompensieren, werden die Vorlesungen 2 volle Zeitstunden dauern.
  - Der daraus resultierende Zeitüberhang wird durch bis zu 4 vorlesungsfreie Termine kompensiert.
  - Bitte informieren Sie sich regelmäßig über diese Termine.
  - **Bisher feststehende Ausfälle:**
    - Bislang keine

Wir haben besonderes Interesse an vorlesungsbegleitendem Feedback, um etwaige Verbesserungsvorschläge ggf. schon während des Semesters zu berücksichtigen.

Übliche Kontaktmöglichkeiten:

- E-mail [jan.jurjens@cs.tu-dortmund.de](mailto:jan.jurjens@cs.tu-dortmund.de)
- Tel.: 0231 755-7208
- Sprechstunde: Montags 15.00-15.50 Uhr (evt. Ausfälle: <http://www-jj.cs.tu-dortmund.de/jj/sprechzeiten.html> )
- Anonymes Kontaktformular: Link von Vorlesungswebseite

Darüberhinaus:

- „1 Minute – 2 Questions“

- Termine (**14 tägig**):
  - Mi. 16:00 bis 17:30, Start: 19.10.2011, OH 14 - 104
  - Do. 10:00 bis 11:30, Start: 20.10.2011, OH 14 - 304
  - Do. 14:00 bis 15:30, Start: 20.10.2011, OH 14 - 104
- Kontakt
  - Bei Fragen zu den Übungen und ihrer Durchführung:
    - Tutor der jeweiligen Gruppe
    - Sebastian Pape:  
[sebastian.pape@cs.tu-dortmund.de](mailto:sebastian.pape@cs.tu-dortmund.de)

VL-Woche	KW	Woche vom	Übung	Gruppe
1	40	10.10.2011	--	--
2	41	17.10.2011	1	1 - 3
3	42	24.10.2011	1	4 - 6
4	43	31.10.2011	2	1 - 3
5	44	07.11.2011	2	4 - 6
6	45	14.11.2011	3	1 - 3
7	46	21.11.2011	3	4 - 6
8	47	28.11.2011	4	1 - 3
9	48	05.12.2011	4	4 - 6
10	49	12.12.2011	5	1 - 3
11	50	19.12.2011	5	4 - 6
12	2	09.01.2012	6	1 - 3
13	3	16.01.2012	6	4 - 6
14	4	23.01.2012	7	1 - 3
15	5	30.01.2012	7	4 - 6

- Anmeldung:
  - Via AsSESS
  - <http://ess.cs.uni-dortmund.de/ASSESS/index.php?do=lecturelist>
  - Anmeldung nach der heutigen Vorlesung möglich
  - Verteilung wird am 17.10.2011 bekannt gegeben

- Übungsmodus
  - Die Übung wird als Präsenzübung durchgeführt.
  - Die Übungszettel werden jeweils 14-tägig Montag morgens veröffentlicht und sind dann Inhalt der nächsten Übung.
  - Die Übungszettel werden während der Übung alleine oder in Gruppen bearbeitet.
  - Der anwesende Tutor steht für Fragen zur Verfügung.
  - Am Ende der Übung werden von den Studierenden Lösungen vorgeschlagen und die Aufgaben besprochen.
  - Ein Lösungsvorschlag zur Präsenzübung wird dann über unsere Webseite veröffentlicht.

- Übungskonzept:

- Insgesamt werden 7 Übungszettel veröffentlicht.
- 6 davon enthalten eine Hausübung.
- Hausübungen sollen bis zum entsprechenden Termin gelöst und abgegeben werden. (Mehr dazu gleich.)
- Die Aufgaben sollen (inhaltlich, konzeptionell) in Gruppen von min. 2 und max. 3 Studierenden bearbeitet werden. Inhaltliche und konzeptionelle Zusammenarbeit sind entsprechend auf den Abgaben zu vermerken.
- Die Abgaben werden korrigiert und die Gruppe erhält die korrigierte Lösung zurück.



- Übungskonzept:
  - Bei jeder Hausübung gibt es 10 Punkte.
  - Diplom-Studierende nach DPO 2001
    - erhalten einen unbenoteten Schein durch erfolgreiche Teilnahme an der Abschlussklausur.
    - Teilnahme an den Übungen und Hausübungen ist freiwillig.
  - Bachelor-Studierende
    - benötigen für die Zulassung zur Klausur einen Leistungsnachweis über die erfolgreiche Teilnahme an den Übungen
    - 50% der Punkte aus den Hausübungen (30 von 60)
    - aber mindestens jeweils 30% der Punkte aus den Aufgaben 1+2, 3+4 und 5+6 (jeweils 6 von 20)

# Übung

## Hausübungen – Zeitschema



◀ Today ▶ October 16 - November 12, 2011 CWS: 42-45 Day Week **Multiweek** Month

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
16 Oct	17	18	19	20	21	22
	Ausgabe		Übung W1	Übung W1		
23	24	25	26	27	28	29
			Übung W2	Übung W2		
30	31 Oct	1 Nov	2	3	4	5
6	7	8	9	10	11	12 Nov
	Abgabe					

# Übung

## Hausübungen – Termine



Übungs-Nr.	Ausgabe	Abgabe
1	17.10.2011	07.11.2011, 10:00 Uhr
2	31.10.2011	21.11.2011, 10:00 Uhr
3	14.11.2011	05.12.2011, 10:00 Uhr
4	28.11.2011	19.12.2011, 10:00 Uhr
5	12.12.2011	16.01.2012, 10:00 Uhr
6	09.01.2012	30.01.2012, 10:00 Uhr

- Ziel: Diskussion der Studierenden untereinander
- Keine Kommunikation mit den Veranstaltern dort
  - Keine garantierten Antwortzeiten
  - Für dringendes: Mail oder Sprechstunde nutzen
- Organisatorische + inhaltliche FAQ
  - Für Fragen von Studierenden, die auch für andere interessant sein könnten
- Moderation durch die Veranstalter

- Prüfung: Klausur
  - schriftlich
  - 120 Minuten
- Klausurtermine:
  - **21.02.12**, 11:00 bis 13:00 Uhr, HS1/HS2, Emil-Figge-Str. 50
  - **26.03.12**, 13:30 bis 15:30 Uhr, HS1 , Emil-Figge-Str. 50

- Bachelor Informatik / Angewandte Informatik:
  - Die Prüfungsleistung wird anhand der Modulprüfung in Form einer schriftlichen Prüfung ermittelt.
  - Die Bearbeitung der Übungsaufgaben ist Voraussetzung zur Teilnahme an der Modulprüfung.



- Jan Jürjens:
  - <http://www-jj.cs.tu-dortmund.de/jj>
- Sebastian Pape:
  - <http://ls14-www.cs.tu-dortmund.de/pape>
- Vorlesungsseite:
  - <http://www-jj.cs.tu-dortmund.de/jj/teaching/ws11/swk/>
- Inpud-Forum:
  - <http://inpud.cs.uni-dortmund.de/phpbb/viewforum.php?f=312>
- Übungsanmeldung
  - <http://ess.cs.uni-dortmund.de/ASSESS>



- Lehrangebot
- Forschung
- Abschlussarbeiten und Hiwi-Jobs



- Professor für Software Engineering an der TU Dortmund
- Wissenschaftskoordinator „Enterprise Engineering“ am ISST
- Leiter der Fraunhofer-Attract-Projektgruppe „Architectures for Auditable Business Process Execution (Apex)“
- Wissenschaftlicher Direktor des EU-Projektes „Security Engineering for Lifelong Evolvable Systems (Secure Change)“
- Senior Member am Robinson College (Univ. Cambridge)

Vorher u.a.:

- Royal Society Industrial Fellow bei Microsoft Research Cambridge
- Research Fellow am Robinson College (Univ. Cambridge)
- Postdoc an der TU München
- Promotion zu „Principles for Secure Systems Design“ (Univ. Oxford)
- Forschungsaufenthalte am LFCS (Univ. Edinburgh) und Bell Labs (Palo Alto)
- Studium an Univ. Bremen und Univ. Cambridge



# Wer ist meine Forschungsgruppe?



- Misha Aizatulin (Microsoft Research Cambridge)
- H. Selcuk Beyhan (Logica (Germany))
- Stephan Braun (TUD)
- Francois Dupressoir (Microsoft Research Cambridge)
- Michael Giddings (Open University)
- Thorsten Humberg (Fraunhofer ISST)
- Christopher McLaughlin (Gartner)
- Martin Ochoa (TUD / Siemens)
- **Sebastian Pape (TUD) => Übungsbetreuung**
- Dr. Thomas Ruhroth (TUD)
- Stefan Taubenberger (Münchener Rückversicherung)
- Daniel Warzecha (Fraunhofer ISST)
- Dr. Sven Wenzel (TUD)
- Christian Wessel (TUD)

IT Systeme durchziehen heute fast alle Funktionen in Wirtschaft und Gesellschaft. IT hat direkten (oft invasiven) Einfluss auf fast alle Aspekte menschlichen Lebens.

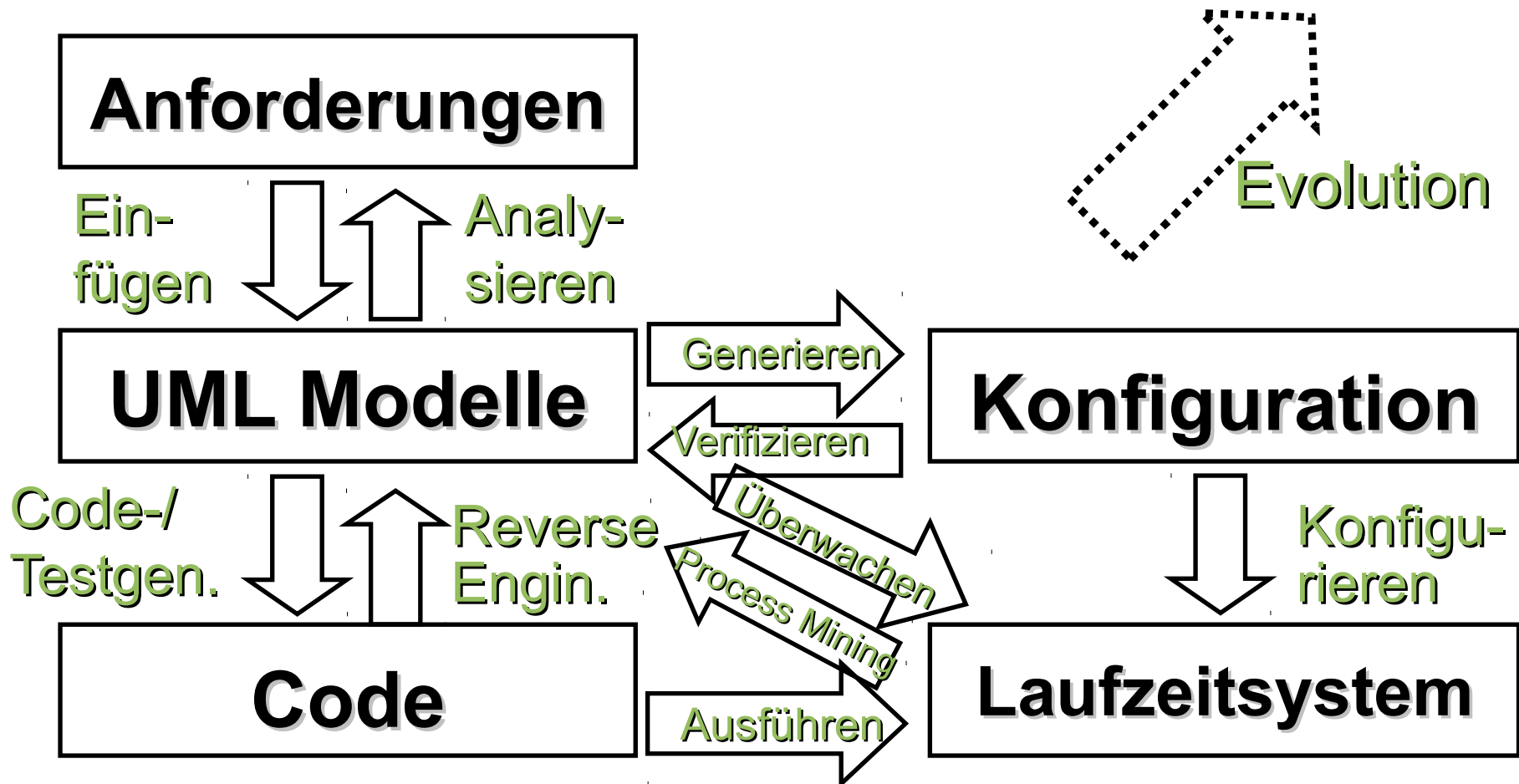
Die Erwartungen an die Vertrauenswürdigkeit dieser Systeme sind daher in den letzten 10 Jahren stark gestiegen. Diese Erwartungen werden oft nicht erfüllt. Teil des Problems ist, dass die bislang verwendeten System- und Software-Entwicklungsmethoden mit den gestiegenen Erwartungen bei gleichzeitig steigender Systemkomplexität nicht mithalten konnten.

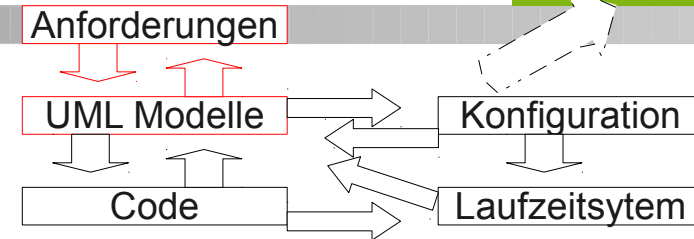
Aus Flexibilitäts- und Kostengründen sind moderne IT Systeme meist über offene Infrastrukturen realisiert, zum Beispiel:

- Internet
- Mobile Netze



Aufgrund ihrer Offenheit sind sie dem Zugriff von Personen ausgesetzt, die in verschiedenem Maße vertrauenswürdig sind. Dieser Zugriff muss daher systemseitig reguliert werden. Aus Flexibilitäts- und Kostengründen wird dies oft auf der Softwareebene gelöst. Eine vertrauenswürdige IT braucht also sichere Software.





## Ziel:

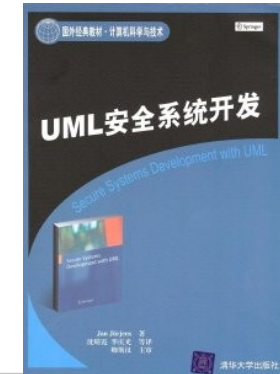
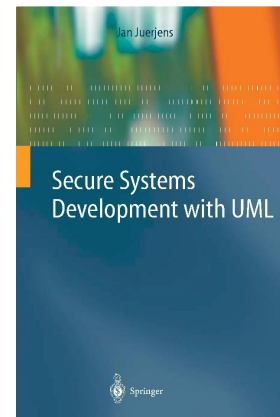
- Dokumentation und automatische Analyse von sicherheits-relevanten Informationen (z.B. Sicherheits-Eigenschaften und -Anforderungen) als Teil der Systemspezifikation.

## Idee:

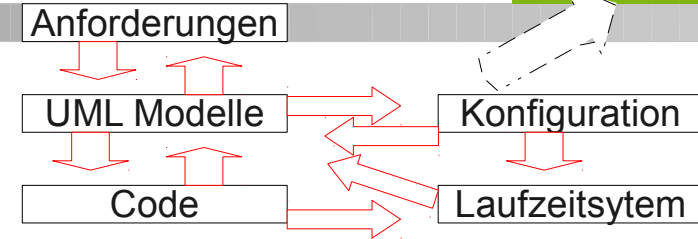
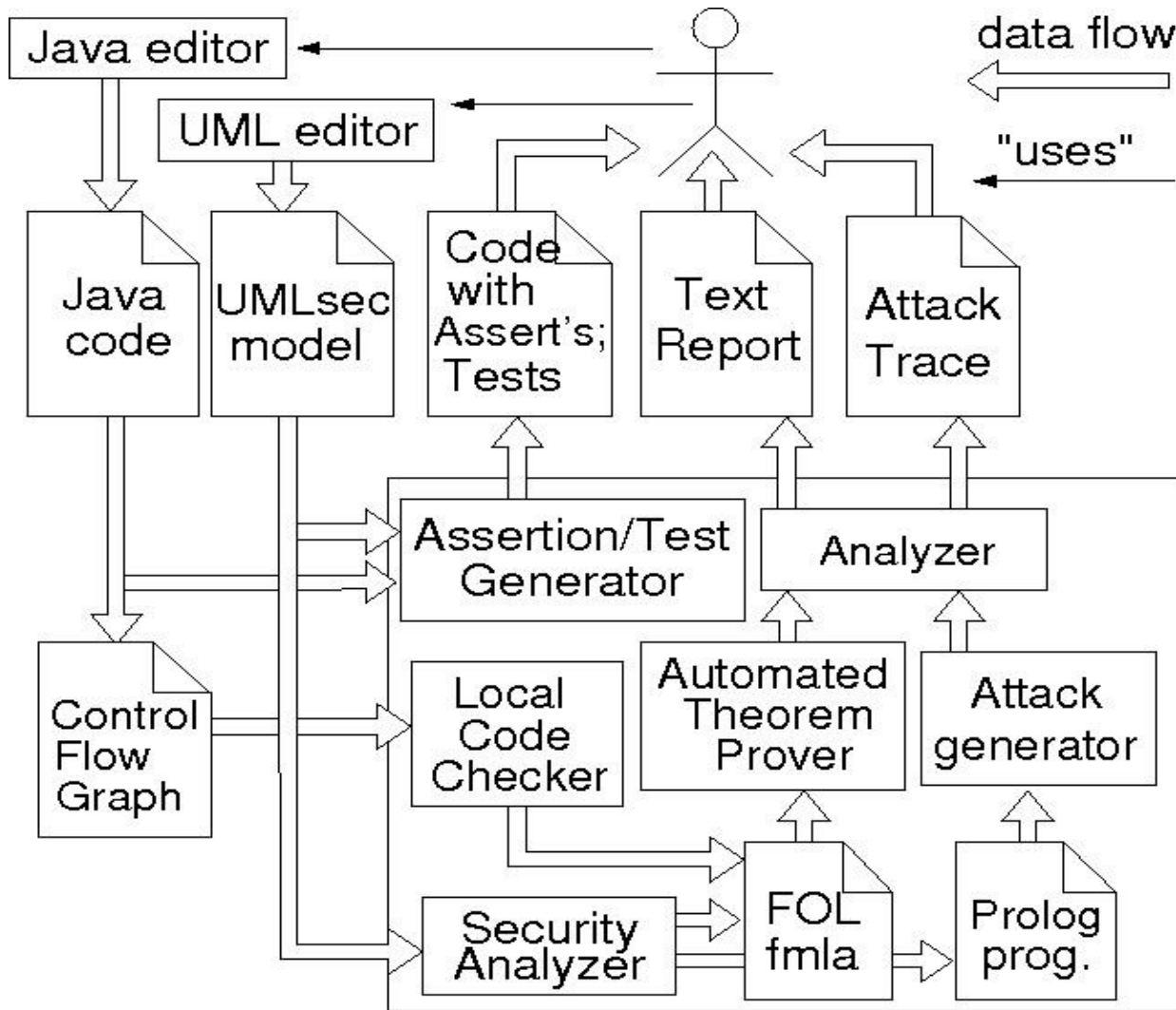
[FASE 01, UML 02]

- UML für System-Modellierung.
- Sicherheitsrelevante Informationen als Markierungen (Stereotypen) einfügen. Definiere dazu UML-Erweiterung UMLsec.
- Formale Semantik mit stromverarbeitenden Funktionen als Grundlage für Verifikation.

[Jour. Logic & Algebr. Program. '08]



# Werkzeugunterstützung



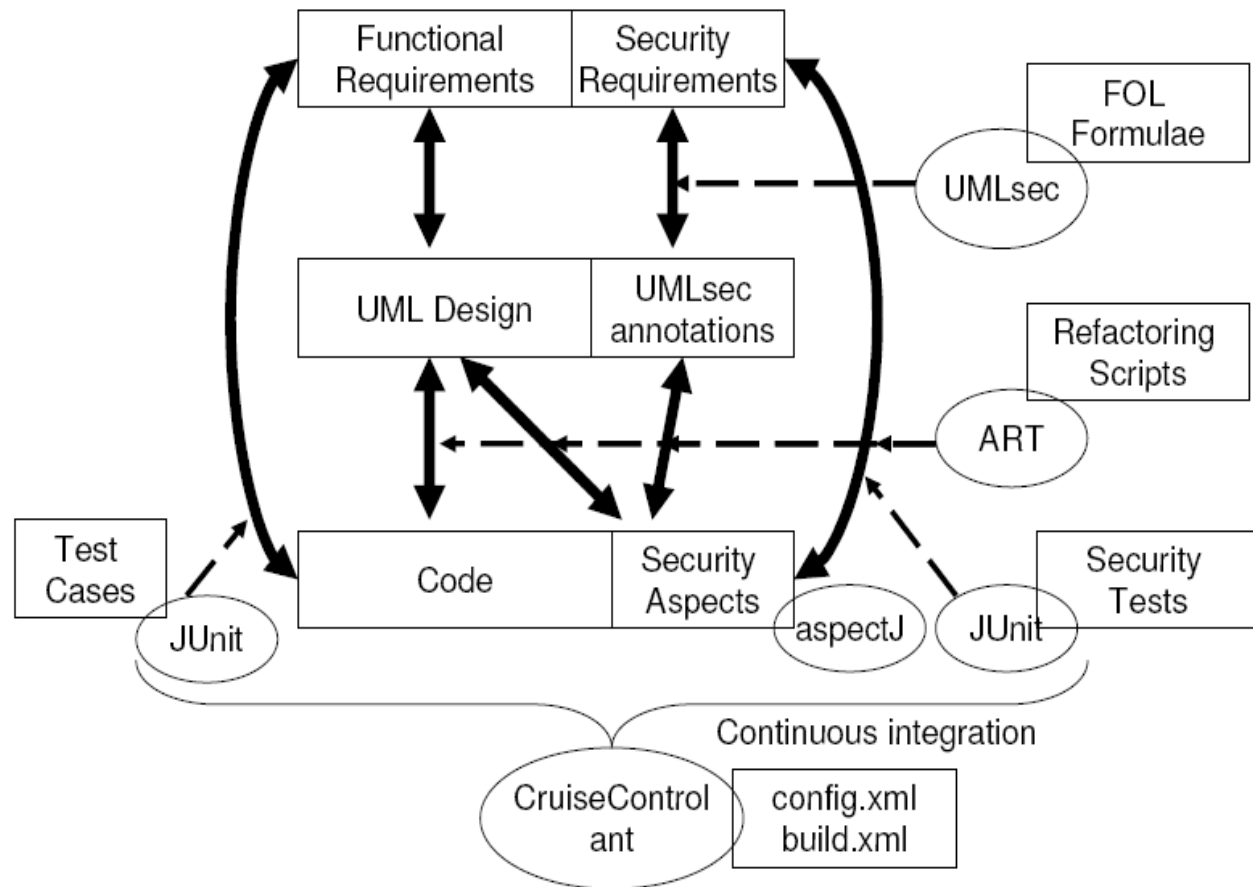
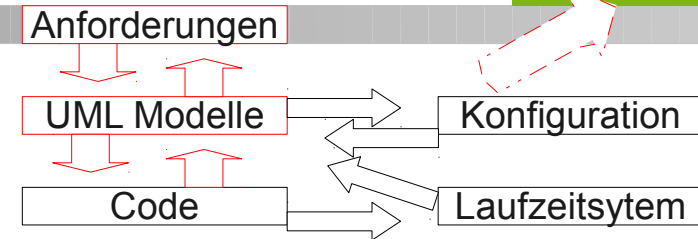
[UML 04, FASE 05,  
Jour. Softw. Tools &  
Techn. Transf.  
(STTT) 07]

# Sichere Evolution: Werkzeugunterstützung



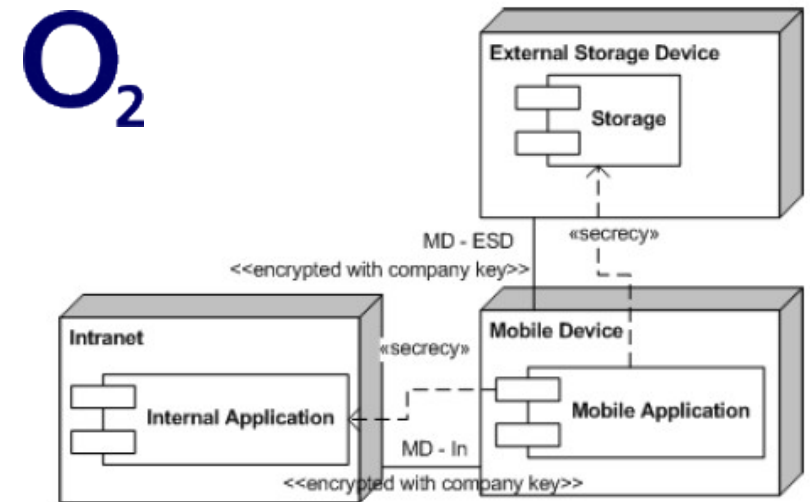
Nachverfolgbarkeit  
von Anforderungen vs. Implementierung  
bei Evolution durch Refactoring  
bewahren.

[ASE'07, ICSM'08, ASE'08,  
Computer Journal '10]



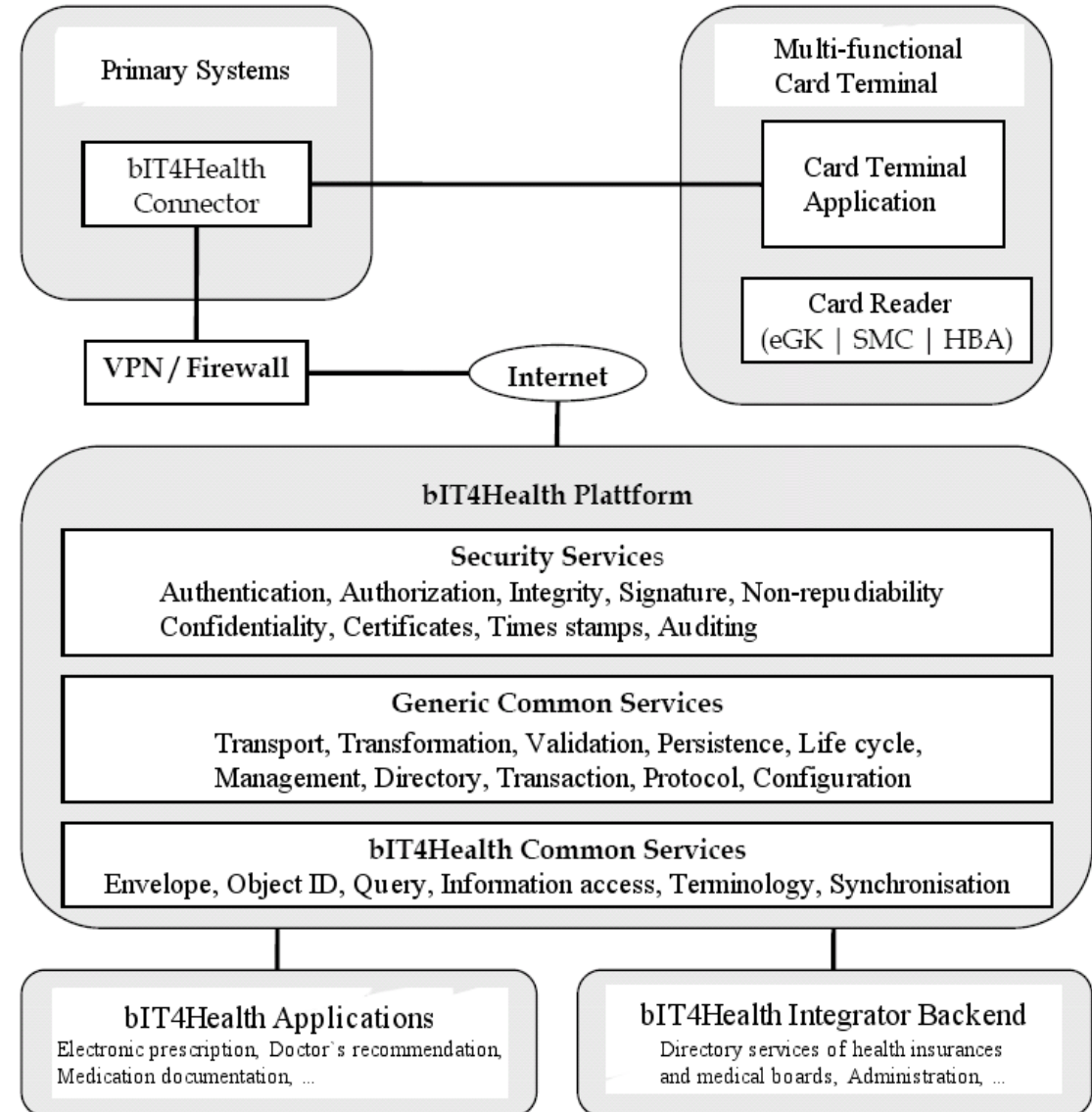


- Anwendung von UMLsec auf mobile Kommunikations-Architekturen bei O<sub>2</sub> (Germany). [ICSE 08]
- Alle 62 Sicherheitsanforderungen aus der Security Policy erfolgreich verifiziert.
- Modellbasierte Techniken bringen Zusatzaufwand.
- Macht sich bezahlt bei wichtigen Sicherheitsanforderungen und Konzentration auf kritische Architekturanteile (auch im Vergleich mit anderen Qualitätssicherungs-Ansätzen mit vergleichbarer Verlässlichkeit)
- UMLsec adäquat für mobile Architekturen.



- Architektur mit UMLsec analysiert.
- Einige Schwachstellen aufgedeckt (fehlender Vertraulichkeitsschutz für digitale Rezepte).

[Meth. Inform.  
Medicine 08]



Modellbasierte Sicherheitsanalyse von webbasierter Bankanwendung (“digitaler Formularschrank”).

Geschichtete Architektur (SSL Protokoll, darauf Client Authentisierungs-Protokoll)

Anforderungen:

- Vertraulichkeit
- Authentisierung

[SAFECOMP 03]



Leben Sie. Wir kümmern uns um die Details.

HypoVereinsbank

Hier empfehlen wir Ihnen mal einen Fonds der Konkurrenz!

**TOOLBOX**

- Lexikon
- Filialfinder
- Formularfinder
- Newsletter
- Geschäftsbedingungen & Konditionen
- Kursuche

- ★ Vorläufiger Konzernabschluss 2001 der HYB Group.
- ★ Die Generation ab 50: Nachlese zum 6. Kompetenz-Kongress.
- ★ "ImmobilienBusiness": das Magazin für Entscheider.
- ★ Die Victoria FörderRente zahlt sich im Alter aus. Lassen Sie sich beraten!
- ★ Zur Guided Tour.

Privatkunden in Sachen Privatleben

Businesskunden In Business-angelegenheiten

Log In Direct B@nking

Direct B@nking Nummer

Kennwort (PIN)

(SSL 3.0) anmelden

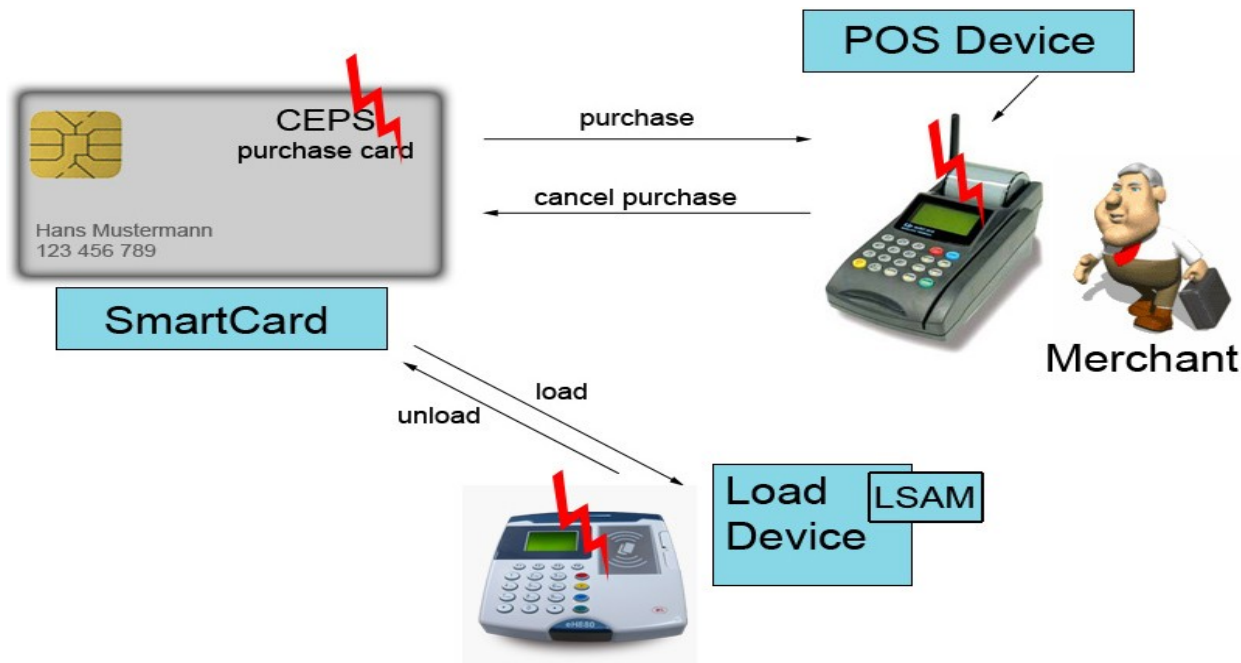
Gastzugang

Common Electronic Purse Specifications:

Globaler Standard für e-Geldbörsen (Visa et al.).

Smartcard enthält Kontostand, sichert Transaktionen mithilfe Krypto.

Formale Analyse von Load und Purchase Protokollen: signifikante Schwachstellen: Kauf-Umleitung, Betrug Ladegerätbetreiber vs. Bank.



[IFIPSEC 01,  
ASE 01]

Smartcard basiertes System.

Analysiert mit UMLsec parallel zur Entwicklung durch Firma in  
gemeinsamem Projekt.

Entdeckten drei signifikante Schwachstellen in verschiedenen  
Versionen (Fehlbedienungsähler umgangen durch Löschen /  
Wiederholen von Nachrichten; Smartcard unzureichend authentisiert  
durch Mischen von Sitzungen).

Endgültig entwickelte Version sicher.

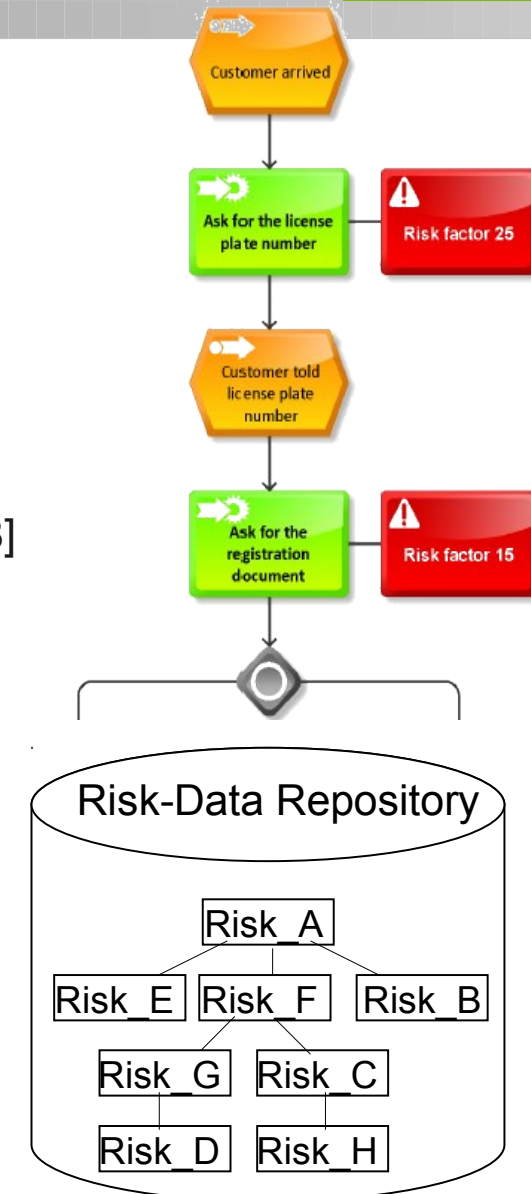
[ACSAC 05]



- Idee: Automatische Analyse von Geschäftsprozessmodellen auf operationale Risiken, z.B. gegenüber Benutzerberechtigungen zur Laufzeit, sowie der Benutzerberechtigungen gegenüber der Sicherheitspolitik,
- automatische Risiko-Identifikation und -Bewertung
- Laufendes Projekt (Fraunhofer Attract): Architekturen für auditierbare Geschäftsausführung (Apex).

[FASE '08]

[ICSE '08]



**BMW Group**

MetaSearch Engine: Personalisierte Suche im Firmen-Intranet (passwort-geschützt).

Einige Dokumente sehr sicherheitskritisch.

Über 1.000 potentielle Benutzer, 280.000 Dokumente, 20.000 Anfragen pro Tag.

Nahtlos in unternehmensweite Sicherheitsarchitektur integriert. Bietet Sicherheitsdienste für Anwendungen (Benutzerauthentisierung, rollenbasierte Zugangskontrolle, globales Single-Sign-On), Ansatzpunkte für weitere Sicherheitsdienste.

Erfolgreich mit UMLsec analysiert.

[ICSE 07]

Es gibt verschiedene Möglichkeiten für eine Beschäftigung als Hiwi am Fraunhofer ISST oder am LS 14 / TUD:

- Unterstützung der folgenden Projekte (beispielsweise durch Java-Programmierung eines UML-Analyse Werkzeuges oder konzeptuelle Arbeiten im Bereich modell-basierte Sicherheitsanalyse):  
"Secure Change", "Architectures for Auditable Business Process Execution (APEX)", „SecureClouds“, „ClouDAT“
- Unterstützung in der Lehre (Tutorien, Folienerstellung etc)

Informationen unter: <http://jan.jurjens.de/jobs/hiwis.html>



Abschlussarbeiten können in inhaltlicher Beziehung zu einer Hiwi-Tätigkeit am Fraunhofer ISST oder LS 14 / TUD durchgeführt werden.

Sie können insbesondere in Zusammenhang mit Anwendungsprojekten am ISST durchgeführt werden, wodurch sich vielfältige Möglichkeiten zu Kooperation mit Unternehmen ergeben, zB:

- Apex: Versicherungen / Banken (Münchener Rückversicherung, Signal Iduna, Wüstenrot), Softwarehersteller (SAP, IDS Scheer)
- Secure Change: Telekom / Smartcards (Telefonica, Gemalto)
- Csec: Microsoft Research Cambridge
- Secure Clouds / ClouDAT: Cloud-Software-Anbieter (LinogistiX), IT-Berater (Admeritia, ITESYS, TÜV-IT)

Informationen unter: <http://jan.jurjens.de/jobs/hiwis.html>

- Formale Abbildung von regulatorischer Compliance auf Security Policies
- Modellierung und Automatische Sicherheits-Analyse für Cloud Computing Systems
- Business Process Mining
- Spezifikation von IT-Sicherheitszielen für die Geschäftsprozessmodellierung und deren Integration in die Ausführung im Workflow
- Design und Entwicklung einer Schnittstelle zwischen der Business Prozess Management Suite ARIS und dem Sicherheitsanalysetool UMLsec zur Compliance Analyse in der Versicherungsdomäne
- Generierung von Geschäftsprozessen mit OpenArchitectureWare unter Berücksichtigung von Sicherheitseigenschaften
- Werkzeuggestützte Modell-basierte Sicherheitsanalyse
- Werkzeugunterstützte Analyse von sicherheitskritischen SAP-Berechtigungen im Finanzbereich
- Modell-basiertes Return on Security Investment (ROSI) im IT-Sicherheitsmanagement

## Dieses Semester:

- Seminar „Ausgewählte Themen des Modell-basierten Sicherheits-Engineerings“.  
<http://www-jj.cs.tu-dortmund.de/jj/teaching/ws11/mbse-sem>  
Bei Interesse bitte bei mir melden.

## SS 2012:

- Methodische Grundlagen des Software Engineering (Master-Basismodul Software) (4+2 SWS)
- FP „Werkzeugunterstützung für UML- und Geschäftsprozessmodelle“ (4 SWS)
- Seminar „Ausgewählte Themen des Modellbasierten Sicherheits-Engineerings“ (2 SWS)

## Zuordnung der Wahlveranstaltungen zu Schwerpunktgebieten (Diplom):

- Sicherheit und Verifikation
- Software-Konstruktion

Forschungsbereich Master: Software, Sicherheit und Verifikation

Informationen unter: <http://jan.jurjens.de/teaching>

Wir haben vielfältige internationale Kontakte, mit denen Auslandsaufenthalte arrangiert werden können, zB:

- Projekt Secure Change: Unis Trento (I), Oslo, Innsbruck, Lille, Besancon, Leuven, Open University (bei London); Firmen Deep Blue (I), Gemalto (F), Telefonica (S), Thales (F)

Viele weitere Kontakte für Auslandsaufenthalte.

“Erfolgreich auch in der Krise“

[<http://fraunhofer.de/presse/presseinformationen/2009/06/Presseinformation18062009Ergebnis.jsp>]

“Mit 1,4 Mrd. Euro erreichte das Finanzvolumen der Fraunhofer-Gesellschaft im vergangenen Jahr ein neues Rekordniveau. ... Im Geschäftsjahr 2008 konnten 1400 neue Stellen besetzt werden. Damit sind 15 000 Mitarbeiterinnen und Mitarbeiter in der Forschungsorganisation tätig. ...

Trotz der weltweiten Finanz- und Wirtschaftskrise geht die Fraunhofer-Gesellschaft für die Jahre 2009 und 2010 von weiterem Wachstum und einer positiven Entwicklung ihrer Ertragslage aus. Der Grund: Zahlreiche Firmen investieren auch in der Krise in Forschung und Entwicklung. ...

Besonders erfreulich: Fraunhofer gehört zu den beliebtesten Arbeitgebern deutscher Studentinnen und Studenten. Das ist das Ergebnis eines Rankings, das die Wirtschaftswoche im Mai diesen Jahres veröffentlicht hat. Laut der Universum Studentenforschung belegt Fraunhofer den 2. Platz nach dem Autobauer Porsche.“

Und: Promotion projekt-begleitend möglich.

Kontakt: <http://jan.jurjens.de>



- Kapitel 1: Intro mit Vorstellung der Professur und Gliederung der Vorlesung
- Kapitel 2: Allgemeine Prinzipien des SW-Engineering
- Kapitel 3: Spezifikation im Allgemeinen
- Kapitel 4: Algebraische Spezifikation
- Kapitel 5: Petrinetze
- Kapitel 6: Modellgetriebene SW-Entwicklung
- Kapitel 7: Object Constraint Language (OCL)
- Kapitel 8: Testen im Allgemeinen, Kontrollflussorientierte Testverfahren, Datenflussorientierte Testverfahren

[Anm.: Teile des Folienmaterials basieren (mit freundlicher Genehmigung) auf Vorlesungsfolien von Prof. Volker Gruhn und Prof. Maritta Heisel.]

- 40 Jahre Software Engineering
- Software Engineering - Wann braucht man's?
- Eigenschaften von Software
- Allgemeine Prinzipien des Software-Engineering



- Allgemeine Modellierungskonzepte, Modellierung
- Was bedeutet Spezifizieren?



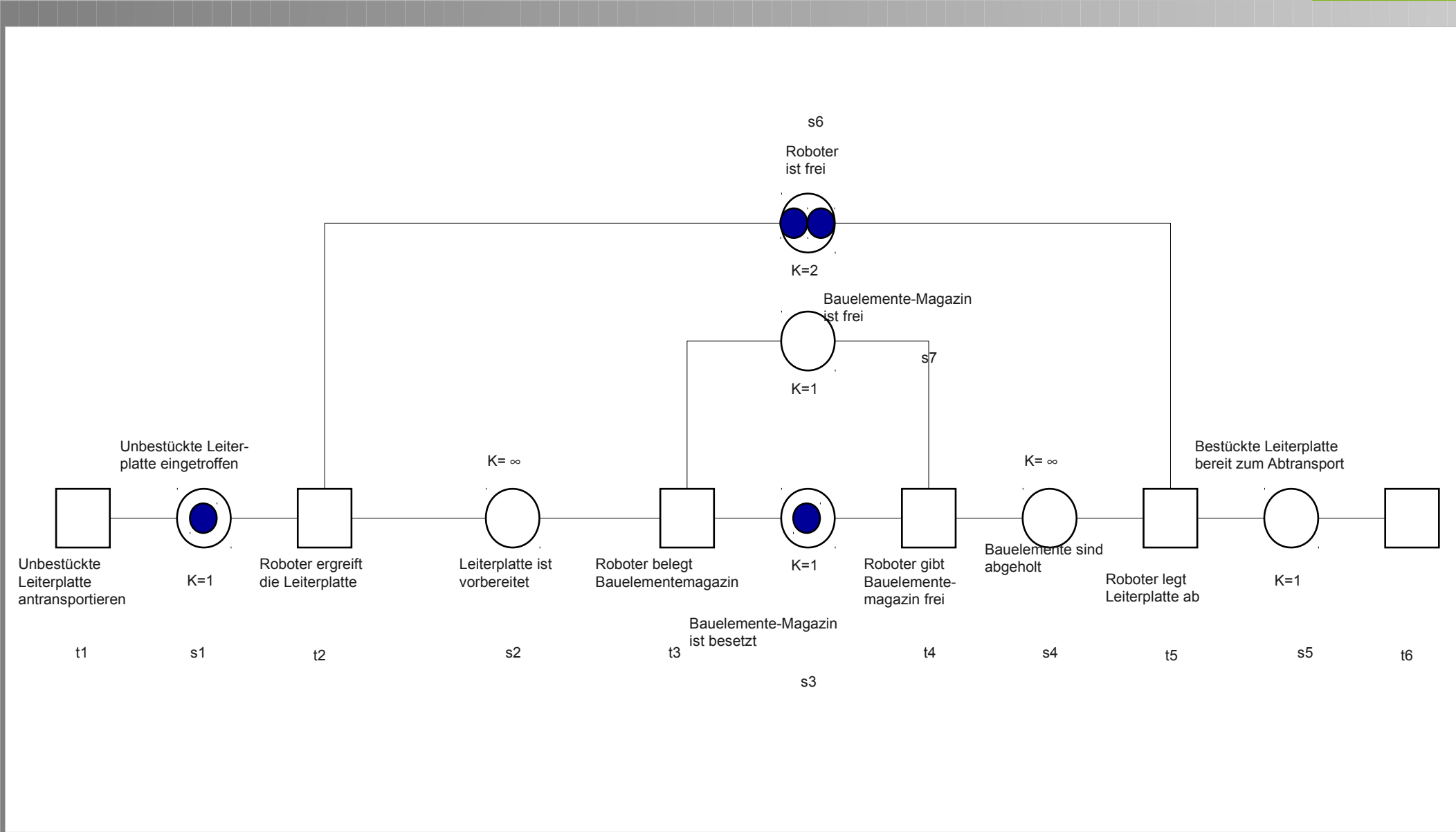
∇  $\Sigma = (S, F)$  heißt algebraische Signatur

- S eine Menge von Sorten
- F eine Menge von Operationssymbolen
- Auf F ist eine Abbildung definiert  
type:  $F \rightarrow S^* \times S$
- Für  $\text{type}(f) = (s_1, \dots, s_n, s)$  schreiben wir  $f: s_1, \dots, s_n \rightarrow s$

∇  $\Sigma$ -Algebra (Definition)

- Seien  $\Sigma = (S, F)$  eine Signatur.
- Für alle  $s \in S$  sei  $A_s$  eine Menge.
- Für alle  $f: s_1, \dots, s_n \rightarrow s \in F$  sei  $f_A: A_{s_1} \times \dots \times A_{s_n} \rightarrow A_s$  eine Abbildung.
- Dann ist das Paar
  - $A = ((A_s)_{s \in S}, (f_A)_{f \in F})$  eine  $\Sigma$ -Algebra

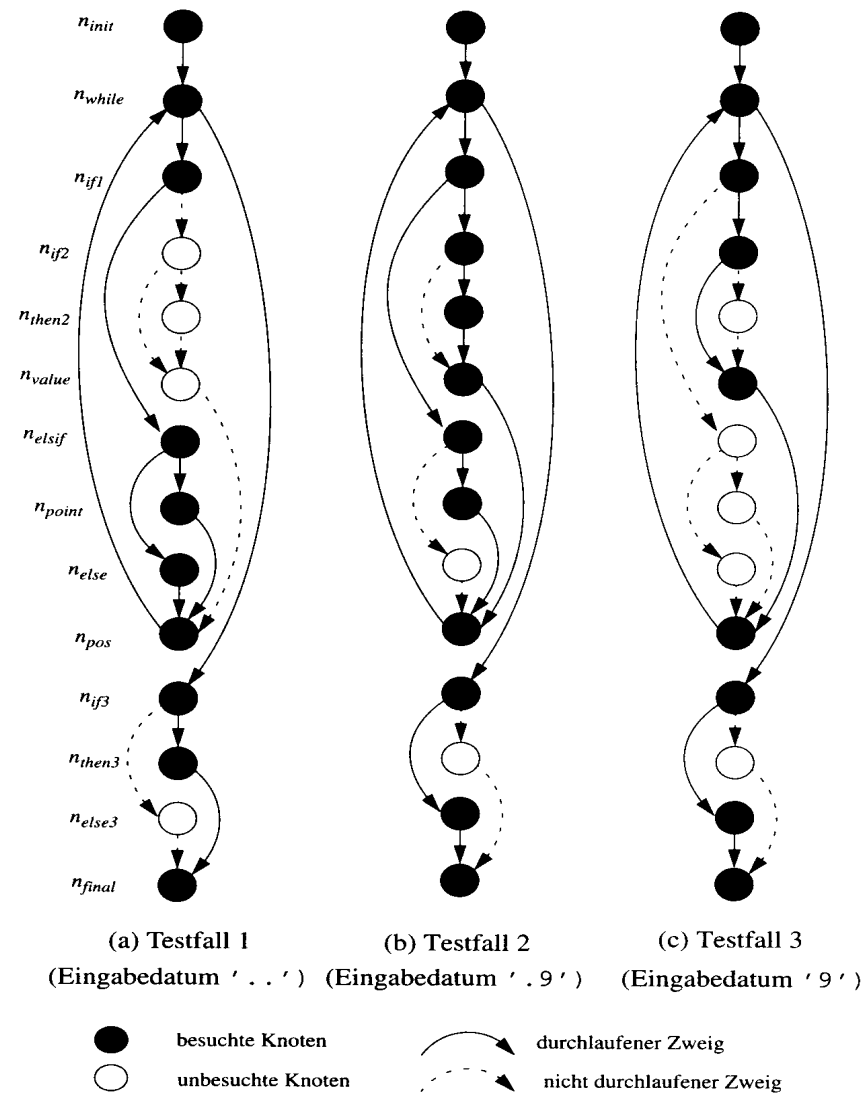
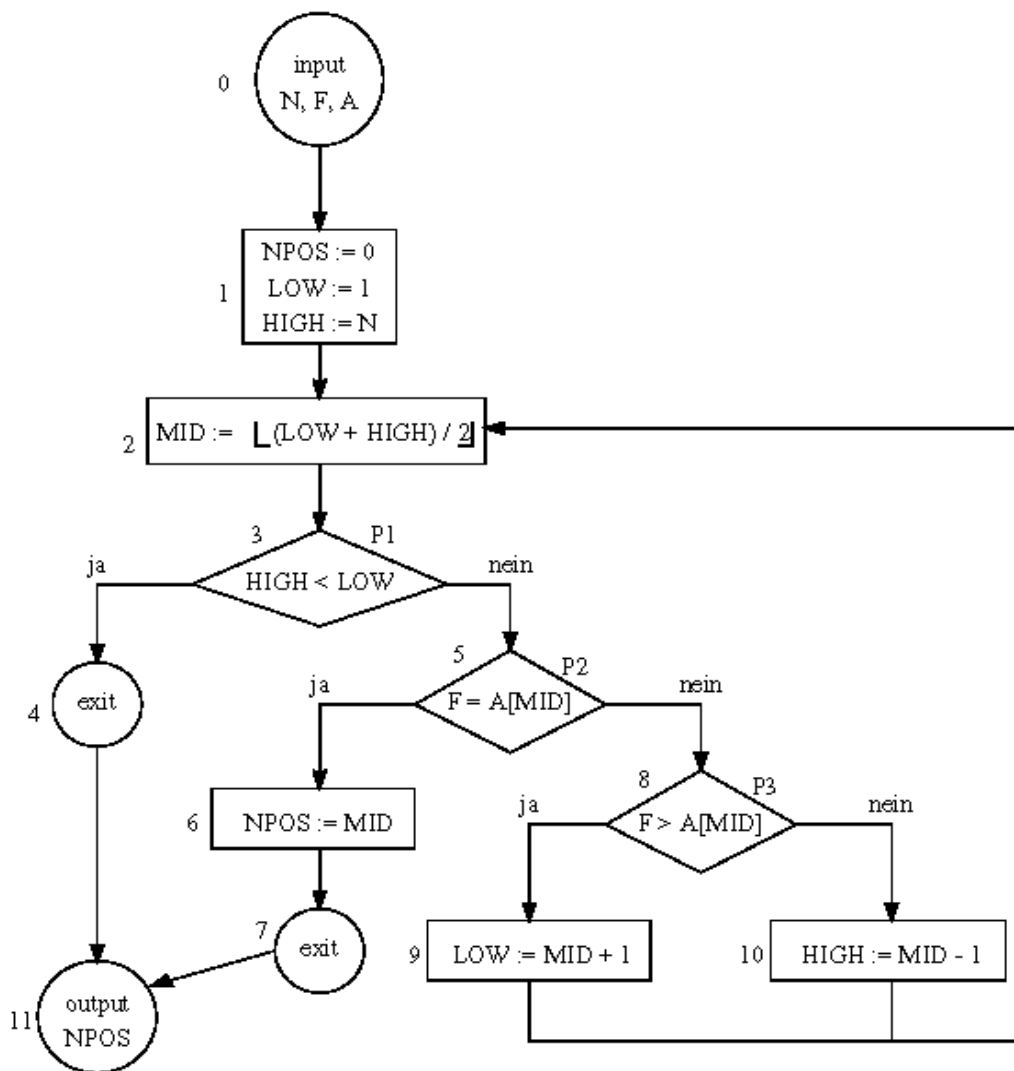
# Kap. 5: Petrinetze



- Modelle und Modellierung
  - Modell-Begriff, Metamodell
  - Domänenspezifische Sprachen
- UML
  - Architektur von UML, MetaModell (Infrastructure)
  - Konzepte der Sprachdefinition (Superstructure)
  - Semantik von UML, Erweiterungen / Tailoring von UML
- Modelltransformationen
  - Allgemeines Konzept der Modelltransformation
  - Verschiedene Konzepte + Transformationssprachen
- Szenarien der Modellgetriebenen SW-Entwicklung
  - Kommunikation mit Modellen, Spezifizieren mit Modellen
  - Testen + Simulieren mit Modellen, Programmieren mit Modellen
  - Berichte aus der MDA-Praxis

Operation	Description
<b>hasReturned() :</b> Boolean	True if type of template parameter is an operation call, and the called operation has returned a value.
<b>result()</b>	Returns the result of the called operation, if type of template parameter is an operation call, and the called operation has returned a value.
<b>isSignalSent() :</b> Boolean	Returns true if the OclMessage represents the sending of a UML Signal.
<b>isOperationCall() :</b> Boolean	Returns true if the OclMessage represents the sending of a UML Operation call.
<b>parameterName</b>	The value of the message parameter.

- Fehler in Softwaresystemen
  - Ursachen, Arten, Kosten und Folgen von SW-Fehlern
  - Ursachen von SW-Fehlern
  - Klassifikation von SW-Fehlern
  - Qualitätsmanagement
- Vorgehensmodelle
- Testverfahren
- Einleitende Begriffe (rund ums Testen)
  - Analysierende Prüfverfahren
  - Reviews, Inspektionen, Walkthroughs
  - statische und dynamische Tests



- C. Ghezzi, M. Jazayeri, D. Mandrioli: Fundamentals of Software Engineering, Prentice-Hall
- Ian Sommerville: Software Engineering, Addison-Wesley
- A.M. Davis: 201 Principles of Software Development, McGraw-Hill
  
- Balzert, H.: Lehrbuch der Softwaretechnik, Band 2, Spektrum, 1998.
- Gao, J. et. Al.: Testing and Quality Assurance for Component-Based Software, Artech House, 2003.
- Meyers, G.: The Art of Software Testing, Wiley&Sons, 1979.
- Riedemann, E.H.: Testmethoden für sequentielle und nebenläufige Software-Systeme, Teubner, 1997
  
- Weitere Literatur in den jeweiligen Vorlesungen

## 1 Zielsetzung der Vorlesung

### 1.1 Ziele

### 1.2 Motivation

- Begriffsbildung
- Relevanz
- Defizite
- Einordnung in die Informatik
- Phänomene des Software Engineerings
- Software-Krise





- Kenntnis der grundsätzlichen verschiedenen Dimensionen des Software Engineerings (technisch, organisatorisch, psychologisch)
- Vermittlung eines Überblicks über das Spektrum gängiger Konzepte zur Spezifikation und zum Testen
- Erläuterung ausgewählter Methoden und Sprachen zur Spezifikation
- Einbettung dieser Spezifikationskonzepte in die Softwaretechnik
- Modellbasiertes Software-Engineering
- Erläuterung verschiedener Testtechniken
- Einbettung in den gesamten Softwareprozess

- Begriffsbildung Software Engineering
  - *The establishment and use of sound engineering principles in order to obtain economically software that is reliable and works efficiently on real machines. [NR68]*
  - *Software Engineering ist die Anwendung wissenschaftlicher Erkenntnisse mit dem Ziel, Computer mittels Programmen, Verfahren und zugehörigen Dokumenten dem Menschen nutzbar zu machen [Den91]*
  - *... [is] the systematic approach to the development, operation, maintenance, and retirement of software.“ [IEEE83]*



- Begriffsbildung Software Engineering ff.
  - *A software engineer must of course be a good programmer, be well-versed in data structures and algorithms, and be fluent in one or more programming languages. ... The software engineer must be familiar with several design approaches, be able to translate vague requirements and desires into precise specifications, and be able to converse with the user of a system in terms of the application rather than in ,computerese‘.*  
[GJM91]
- Daraus folgende, notwendige Fähigkeiten:
  - Kommunikation auf verschiedenen Abstraktionsebenen
  - Erstellung und Verwendung von Modellen / Spezifikationen
  - Kommunikation mit Personen mit unterschiedlichen Zielsetzungen, Vorstellungen, Ausbildungen
  - Arbeitsplanung und -koordination

- Defizite des Software Engineerings
  - Komplexitätsbeherrschung
  - Überbewertung des Formalisierbaren
  - Unterbewertung der Anschauung
  - Unterschätzung verfahrensorientierter Arbeitsteilung (bei gleichzeitiger Überschätzung der formalisierbaren Arbeitsteilung)
  - Ausbildungsdefizite
  - „Over-Engineering“ (übertriebene technische Komplexität)

- Phänomen der Software-Entwicklung
  - Trotz der Abhängigkeit von Software gibt es
    - Keine zuverlässige Herstellung von Software im industriellen Maßstab
    - Kosten- und Terminüberschreitungen
    - Bei Auslieferung: ungenügende Softwarereife
    - Keine Produktivitätskontrolle wie in anderen industriellen Fertigungsbereichen
    - Keine Qualitätskontrolle wie in anderen industriellen Fertigungsbereichen, gerade das Testen ist immer noch unterbewertet.

- Ursachen der Phänomene (nach [CKI88])
  - Unzureichende Verbreitung von Anwendungsdomänen-Wissen
    - Projekte in neuer Branche
    - Technologiezentriertheit
  - Ändernde und sich widersprechende Anforderungen
    - Markt
    - verschiedene Kunden
    - Erkenntnisprozesse
    - Missverständnisse
  - Kommunikations- und Koordinierungs-Pannen
    - unterschiedliche Vorstellungen und Zielsetzungen
    - Inkongruenzen zwischen Kompetenz und Verantwortung
    - Kapitulation

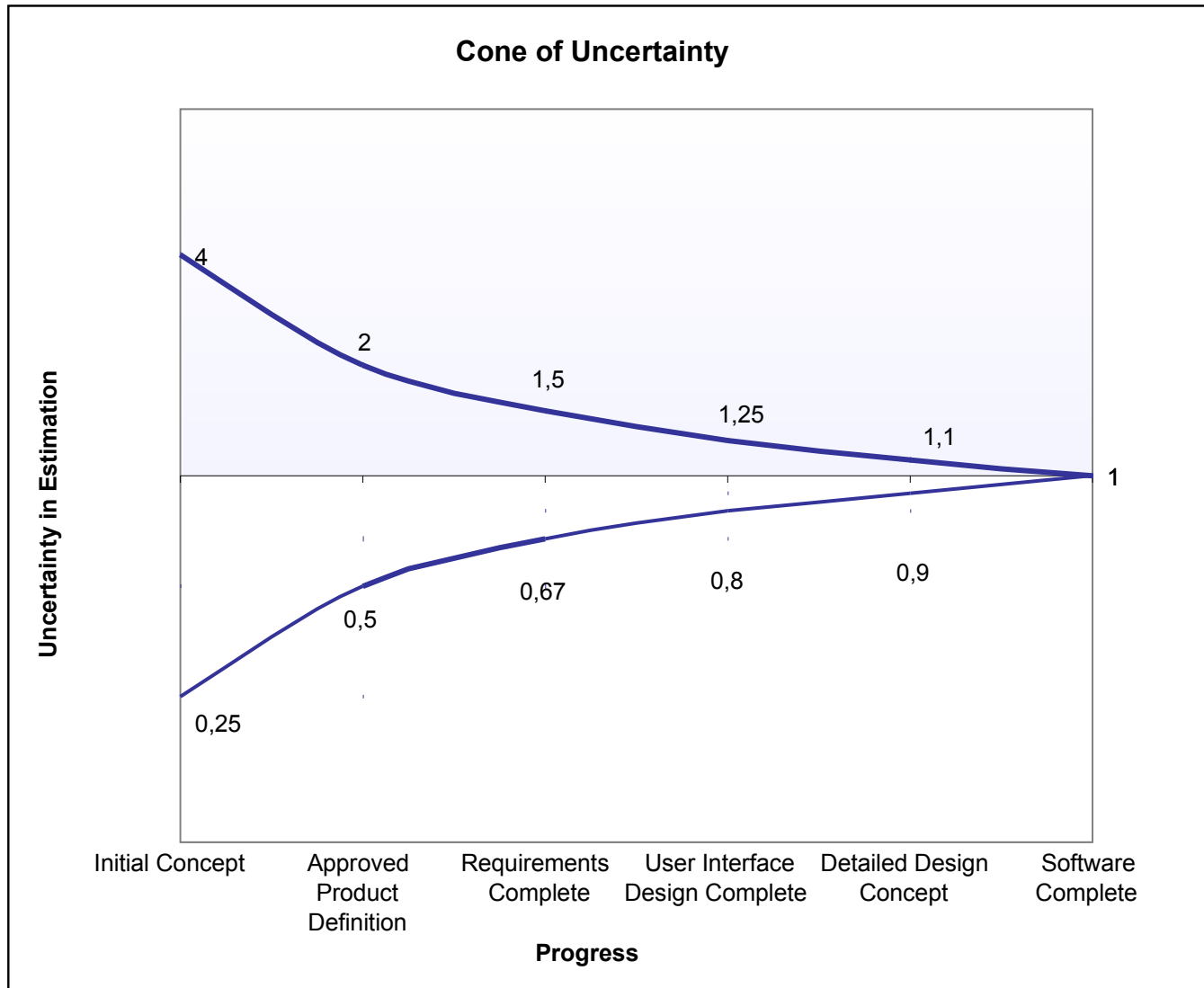


- Phänomen I: Abbruchrate

Anzahl Function	Früher als geplant	Termingerecht	Verspätet	Abgebrochen
1 FP	14,86%	83,16%	1,92%	0,25%
10 FP	11,08%	81,25%	5,67%	2,00%
100 FP	6,06%	74,77%	11,83%	7,33%
1.000 FP	1,24%	60,76%	17,67%	20,33%
10.000 FP	0,14%	28,03%	23,83%	48,00%
100.000 FP	0,00%	13,67%	21,33%	65,00%
<b>Durchschnitt</b>	<b>5,53%</b>	<b>56,94%</b>	<b>13,71%</b>	<b>23,82%</b>

Abbruchrate großer Projekte nach [Jon96]

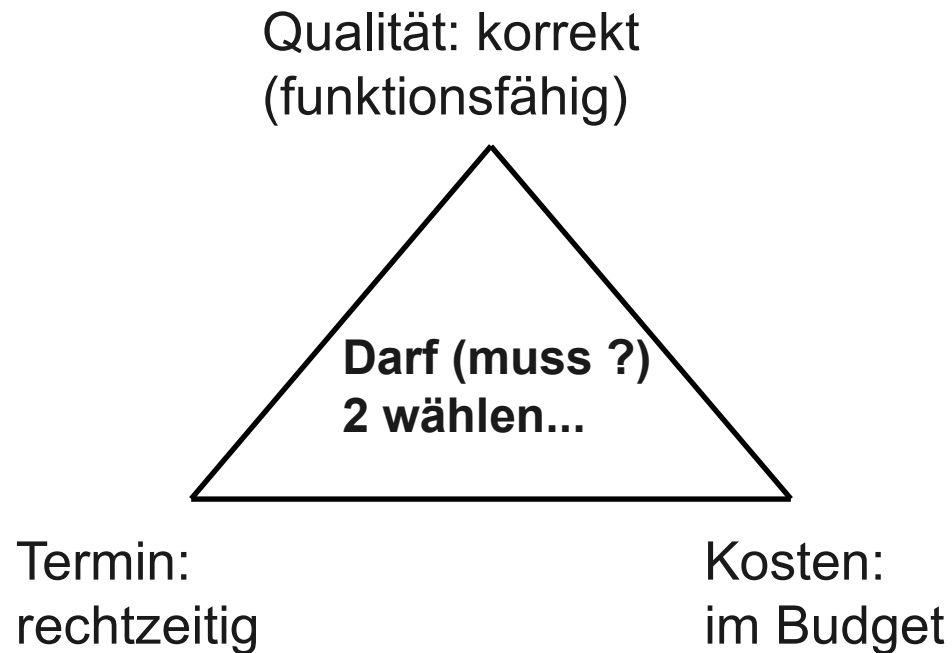
1 Function Point (FP) sind im Schnitt 55 Lines of Code (LOC) in C++/Java, 20 LOC Perl, 13 LOC SQL, etc.  
 Desktop-Projekt mit 1 FP dauert ca. eine Woche mit einem Entwickler  
 Allgemeines Projekt mit 100.000 FP und 100 Entwicklern: ca. 17 Monate  
 (inkl. Planung, Entwurf, Test etc.; ohne Urlaub, Krankheiten etc.)  
 Formel z.B. für Desktop Projekt:  $StaffMonth = 0,157 * FunktionPoints^{0,591} * MaximumTeamSize^{0,810}$



Nach: Steve McConnell: Software Estimation, Microsoft Press, Redmond, Washington, 2006.



- Phänomen II: Termine, Kosten, Qualität



NR68, P. Naur, B. Randell (eds), Software Engineering: Report on a Conference, NATO Scientific Affairs Division, Brüssel, 1968

Den91, E. Denert, Software-Engineering, Berlin, Springer, 1991

IEE83, IEEE Standard Glossar of Software Engineering terminology - IEEE Standard 729, IEEE Computer Society Press, New York, 1983

CKI88, B. Curtis, H. Krasner, N. Iscoe, A Field Study of the Software Design Process for Large Systems, Communications of the ACM, November 1988, Vol. 31, No. 11, pp 1268 – 1287

Jon96, C. Jones, Large Software System Failures and Successes, in: American Programmer, Vol. 9, No. 5, September 1996