

Software Engineering

The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software.

IEEE Std. 610.12 (1990)

Methoden des Software Engineering

Prof. Dr. Christian Prehofer

Unter Verwendung von Materialien von
Nora Koch und Martin Wirsing

WS 2012/13

Vorlesung: Methoden des Software Engineering
Block A

Einführung und Übersicht

Christian Prehofer

15.10.2012

4. Juni 1996: Erster Start der "Ariane-5"



- Während des Flugs läuft ein unnötiges Kalibrierungsprogramm für die Trägheitssensoren. Die gemessenen Werte der Ariane-5 überschreiten die in der Ariane-4-Software vorgesehenen Bereiche.
- Die dadurch ausgelöste (Ada-)Exception wird durch Anhalten des Steuerungscomputers behandelt, um auf ein zweites redundantes System umzuschalten.
- Im zweiten System tritt der gleiche Software-Fehler auf und wird identisch behandelt.
- Kosten des Ariane-5-Programms bis 1996 *ca. 7 Milliarden US-\$*
- Wert des zerstörten Satelliten: *ca. 500 Millionen US-\$*

Software-Katastrophe: Kein Einzelfall

■ Technik-Katastrophen:

- September 1999: Verlust der Sonde "Mars Climate Orbiter" wegen falscher Einheitenrechnung
- 1985-1987 Therac 25 (Strahlengerät zur Krebsbehandlung): Fehlerhafte Programmierung führt zu Verbrennungen und Todesfällen

■ Finanzielle Katastrophen:

- 1990 AT&T Telefonverbindung zwischen Ost- und Westküste der USA wg eines SW-Fehlers für mehr als 24 Std unterbrochen: *ca. 1 Mia US-\$*
- 1992: Integration des Reservierungssystems SABRE mit anderen Reservierungssystemen abgebrochen: *165 Mio. US-\$*

■ Terminkatastrophen:

- 1994: Eröffnung des Denver International Airport um 9 Monate verzögert wegen Softwareproblemen im Gepäcktransport-System
- 2003: Einführung des LKW-Mautsystems in Deutschland verzögert sich um 18 Monate

■ Ausführliche Liste von Software Bugs und deren Auswirkungen

- <http://wwwzenger.informatik.tu-muenchen.de/persons/huckle/bugse.html>

Permanente Software-Krise?

- **1965: Der Begriff der *Softwarekrise* etabliert sich in Industrie und Wissenschaft.**
 - Fehler in Computersystemen sind fast immer auf Softwarefehler zurückzuführen
 - Software wird nicht termingerecht und/oder zu höheren Kosten als geschätzt fertiggestellt
 - Software entspricht oft nicht den Anforderungen ihrer Benutzer
- **Studie von 1979 zu Softwareprojekten (USA):**
 - 75% der Ergebnisse nie eingesetzt
 - 19% der Ergebnisse stark überarbeitet
 - 6% benutzbar.
- **Studie von 1994 zu Software-Großprojekten (IBM Consulting):**
 - 55% Kostenüberschreitung
 - 68% Terminüberschreitung
 - 88% Bedarf für starke Überarbeitung

Was ist Software Engineering?

- Eine naive Sicht:



- Aber ...

- Wie **spezifiziere** ich das Problem ?
- Woher weiß man, dass die Spezifikation die **Nutzerbedürfnisse** erfüllt?
- Wie wurde die **Struktur des Programms** bestimmt?
- Woher weiß man, dass das Programm die **Spezifikation erfüllt**?
- Woher weiß man, dass das Programm **immer zuverlässig arbeiten** wird?
- Wie organisiere ich die Aufgaben, wenn ein **Team von Entwicklern** zur Verfügung steht ?

Besonderheiten von Software

- Software ist **immateriell**.
- Software unterliegt **keinem Verschleiß**.
- Es gibt **keine Software-Ersatzteile**:

Defekte sind immer Konstruktionsfehler.

- Software ist **schwer zu vermessen** („Technische Daten“ von Software?).
- Software gilt als relativ **leicht änderbar**
(im Vergleich zu materiellen technischen Produkten).
- Software unterliegt einem **ständigen Anpassungsdruck**.
- Software **altert**.

Software is a hybrid, halfway between an abstract idea and a physical, tangible thing. Software is neither land nor sea, but swamp: a hybrid too thin for the army (software engineering) and too thick for the navy (computer science).

Brad Cox

Software Engineering (I)

Einige Definitionen und Aspekte

- *The establishment and use of sound engineering principles in order to obtain economically software that is reliable and runs on real machines.
F.L. Bauer, NATO-Konferenz Software-Engineering 1968*
- *The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software.*

IEEE Std. 610.12 (1990)

Also:

- Bereitstellung und systematische Anwendung von Methoden, Verfahren und Werkzeugen zur Entwicklung, Betrieb und Wartung von Software.

Software Engineering (II)

“state of the art of developing quality software on time and within budget”

- Trade-off zwischen Perfektion und physischen Grenzen
 - SE muss sich mit Problemen der realen Welt auseinandersetzen
- Stand der Kunst!
 - Die Gemeinschaft entscheidet über „Beste Praxis“
 - Lebenslange Weiterbildung

Software Engineering (III)

“multi-person construction of multi-version software”

David L.Parnas

- Teamarbeit
 - Aspekt der Größenordnung (“Gut programmieren” ist nicht genug)
 - Kommunikationsaspekt
- Erfolgreiche Softwaresysteme müssen sich weiterentwickeln oder sie werden verschwinden

Änderung ist die Norm, nicht die Ausnahme

Software-Produktqualität (ISO 9126/DIN 66272)

- **Funktionalität**
 - Angemessenheit
 - Sicherheit
 - Genauigkeit der Berechnung
 - Interoperabilität
 - Konformanz zu Standards
- **Zuverlässigkeit**
 - Reife
 - Fehlertoleranz
 - Wiederherstellbarkeit
- **Benutzbarkeit**
 - Verständlichkeit
 - Erlernbarkeit
 - Bedienbarkeit
- **Effizienz**
 - Zeitverhalten
 - Verbrauchsverhalten
- **Änderbarkeit**
 - Analysierbarkeit
 - Modifizierbarkeit
 - Stabilität
 - Prüfbarkeit
- **Übertragbarkeit**
 - Anpassbarkeit
 - Installierbarkeit
 - Konformanz zu Standards
 - Austauschbarkeit

Anforderungen für Qualitätssicherung

- Sicherheitsstandard ISO WD 26262 (2009-2011)
 - Sicherheitskritische Systeme sind nach dem Stand der Technik zu entwickeln. Anderfalls drohen Haftungsansprüche ggü. den verantwortlichen Entwicklern.
 - Für alle Phasen des Produktlebenszyklus legen sie in Abhängigkeit von der Kritikalität des Systems Rahmenbedingungen für die durchzuführenden Prozesse fest.

- Kritikalitätsstufen in ISO WD 26262:
 - ASIL A-D (Automotive Safety Integrity Level)
 - Einstufung hängt ab von potentieller Unfallschwere, Eintretenswahrscheinlichkeit und Beherrschbarkeit bei einer Fehlfunktion.
 - ASIL A = geringste Stufe, ASIL D = höchste Stufe

Methoden des Software Engineering: WS 2011/12

Dozenten:

Prof. Dr. Christian Prehofer

ECTS-Punkte: 6

Folien normal am Vorabend der Vorlesung im Netz

Prüfungen / Note

- 4 Kurztests in den Übungen, je ca. 15min
 - Die besten 3 davon zählen 30%
- Klausur, 70%,

Klausurtermin

- In Vorbereitung

Übungen zur Vorlesung

Voraussetzungen:

- [Anmeldung in Uniworx](#)

Übungsbetrieb: Wöchentliche Übungen ab 16.10

- Dienstag 9-12 Uhr, Richard-Wagner-Strasse 10, Raum 110
- Hinweis: Exkursionen / Industrievorträge geplant (s. Folie 16)



Literatur

- Ian Sommerville: Software Engineering, 9th edition, Addison-Wesley 2010.
- Helmut Balzert: Lehrbuch der Software-Technik (3 Bände), Spektrum Akademischer
 - Lehrbuch der Softwaretechnik: Basiskonzepte und Requirements Engineering, 2009
 - Lehrbuch der Softwaretechnik: Softwaremanagement, 2008
 - Lehrbuch der Softwaretechnik: Entwurf, Implementierung, Installation und Betrieb (2011)
- Leider decken diese (Standard-)Bücher die Vorlesung nicht komplett ab
 - V.a. Agile Methoden, Formale Verifikation
- Weitere Literatur bei den einzelnen Modulen

Vorlesungsüberblick

| Themenblock | Thema |
|-----------------------------|---------------------------------------|
| A) Einführung | Einführung, Überblick |
| B) Prozeß | Wasserfall, V-Modell, Iterativ |
| | Agile Methoden / Scrum |
| | Projektmanagement |
| C) Anforderungen | Anforderungsanalyse |
| | Zielorientierte Methoden |
| | Anforderungsspezifikation |
| D) Architektur | Systemarchitektur |
| | Muster |
| E) Test, Validierung | Qualitätsmanagement, SW-Test |
| | Black-Box, White-Box test |
| F) Formale Methoden | Einführung Formale Methoden |
| | Modellprüfung |

Geplante Industriebeiträge

- 30.10: Aufwandsschätzung für Software
 - Exkursion zur Firma MSG
- 6.11: Projektmanagement
 - Vortrag Firma Accenture
- Qualitätssicherung (in Planung)
 - Firma BMW
- Architecture Review (in Planung)
 - Firma Siemens

Exkursion zur Firma MSG

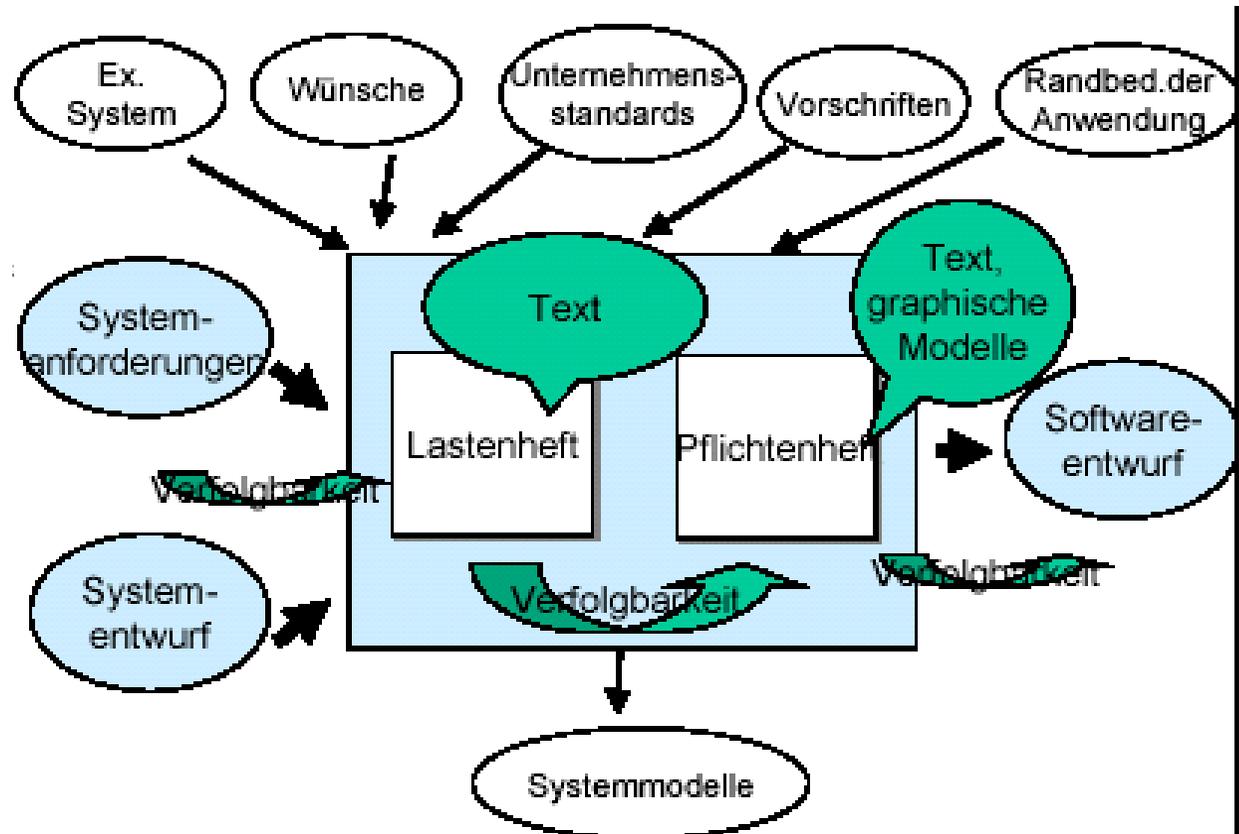
Aufwandsschätzung für Software

- **Termin: Dienstag, 30. Oktober 2012, 09:15 – 13:15 Uhr**
- **Ort:** msg systems ag, Robert-Bürkle-Str. 1, 85737 Ismaning
- **Teilnehmer:** Anmeldung über C. Kroiss/Uniworks;
 - **Laptops mitbringen**
- **Programm**
 - 09:15 – 09:45 Begrüßung & Vorstellung msg systems ag
 - 09:45 – 11:15 **Aufwandsschätzung** in der industriellen Praxis
 - 11:15 – 13:15 **Fallstudie** Aufwandsschätzung mit msg Werkzeugen am „echten Case“
 - Optionales Programm der Firma MSG
 - 13:15 – 14:00 - Mittagessen mit Mitarbeitern der msg -
 - 14:00 – 14:30 „Berufseinstieg als Informatiker“ – ein persönlicher **Erfahrungsbericht**
 - 14:30 – 15:00 **FAQ** – Berufsbild Informatiker, am Beispiel der msg

Übersicht über die Vorlesung

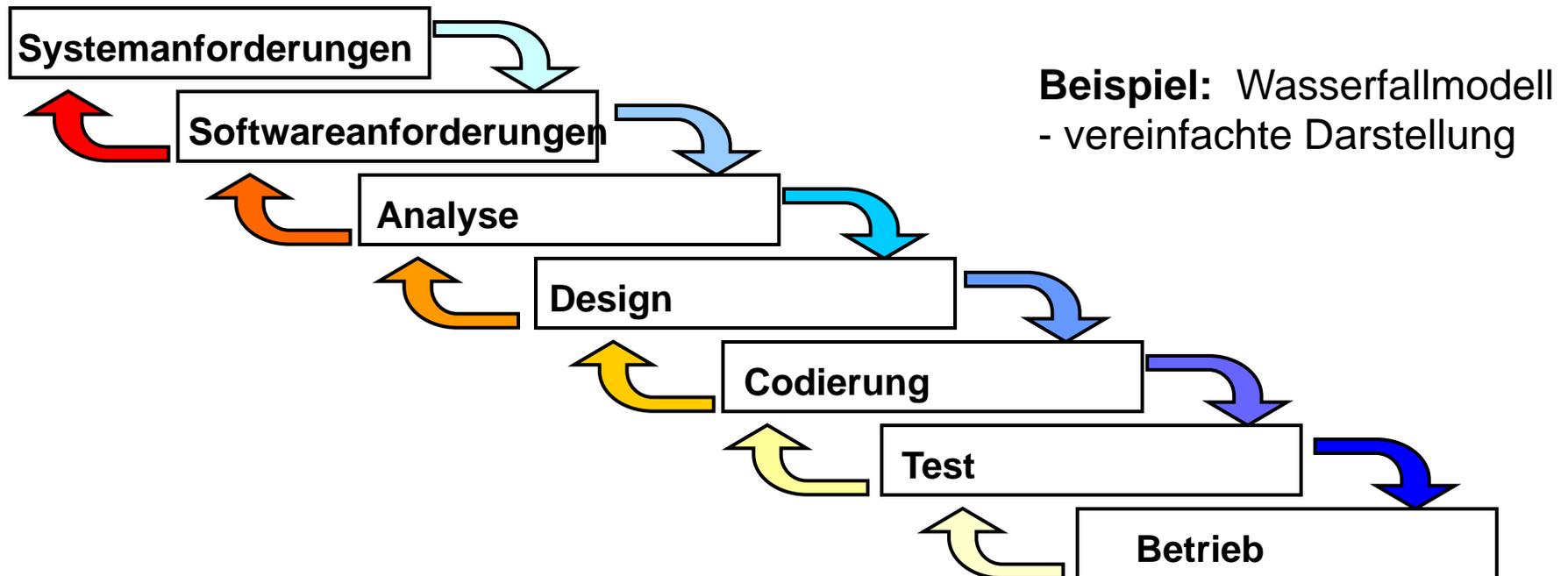
Anforderungsanalyse

- Anforderungserwerb
- Konzeptuelle Modellierung
- Anforderungsvalidierung



Prozess und Projektmanagement

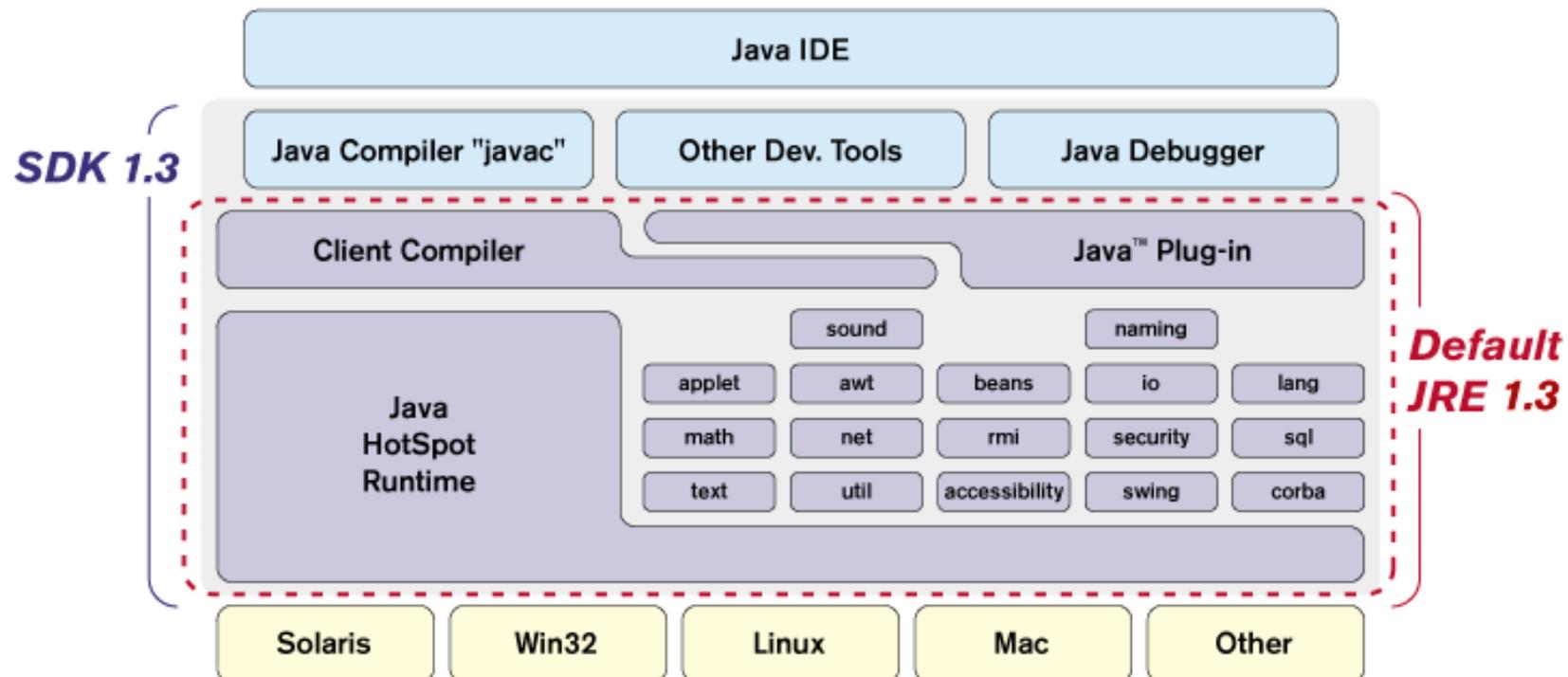
- Vorgehensmodelle
- Prozessverbesserung
- Projektmanagement
- Empirisches Software-Engineering



Software-Architektur

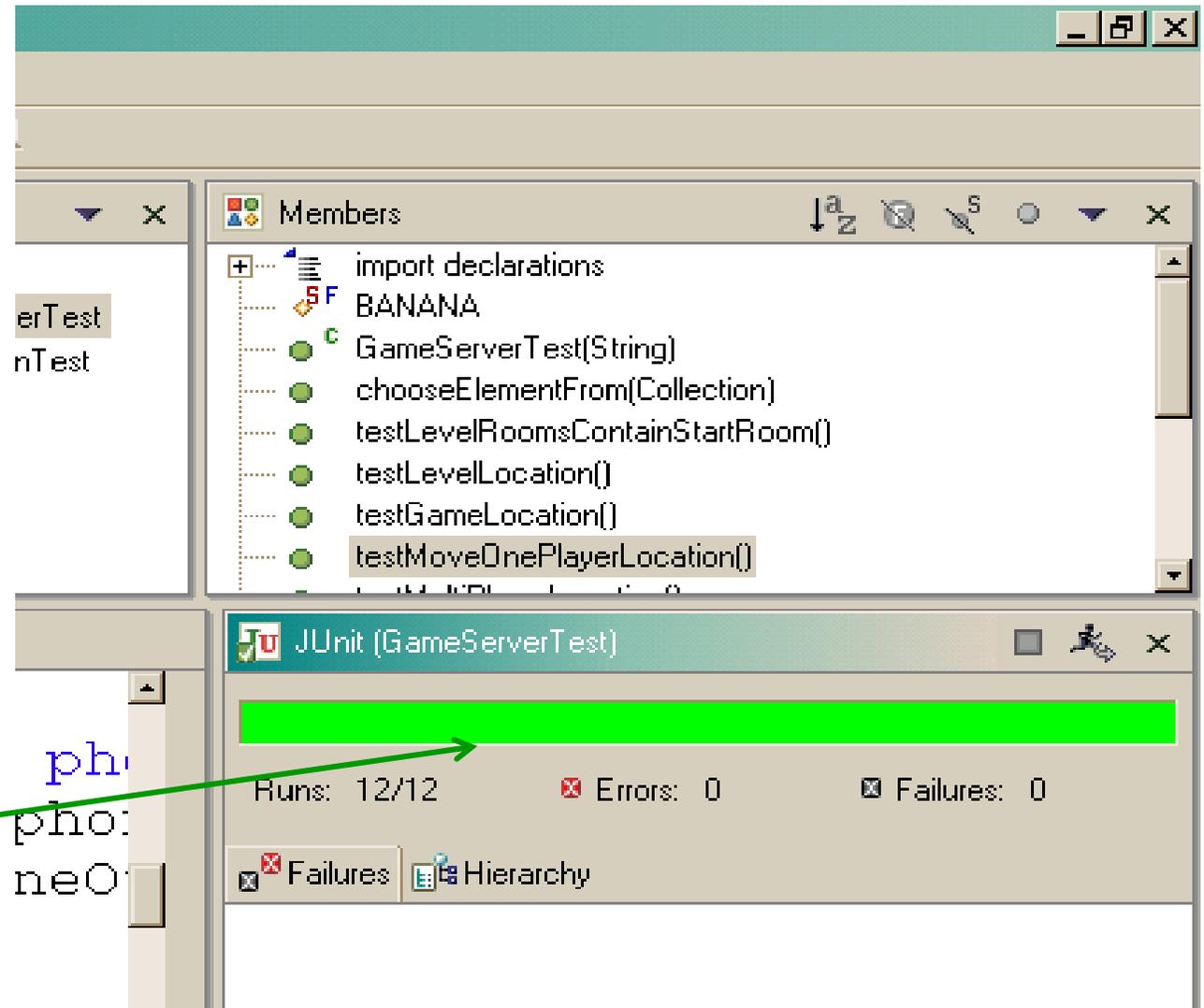
- Architekturentwurf
- Systemarchitektur
- Middleware
- Entwurfsmuster & Komponenten

Beispiel:
Architektur
einer Java-
Entwicklungs-
plattform
(informell)



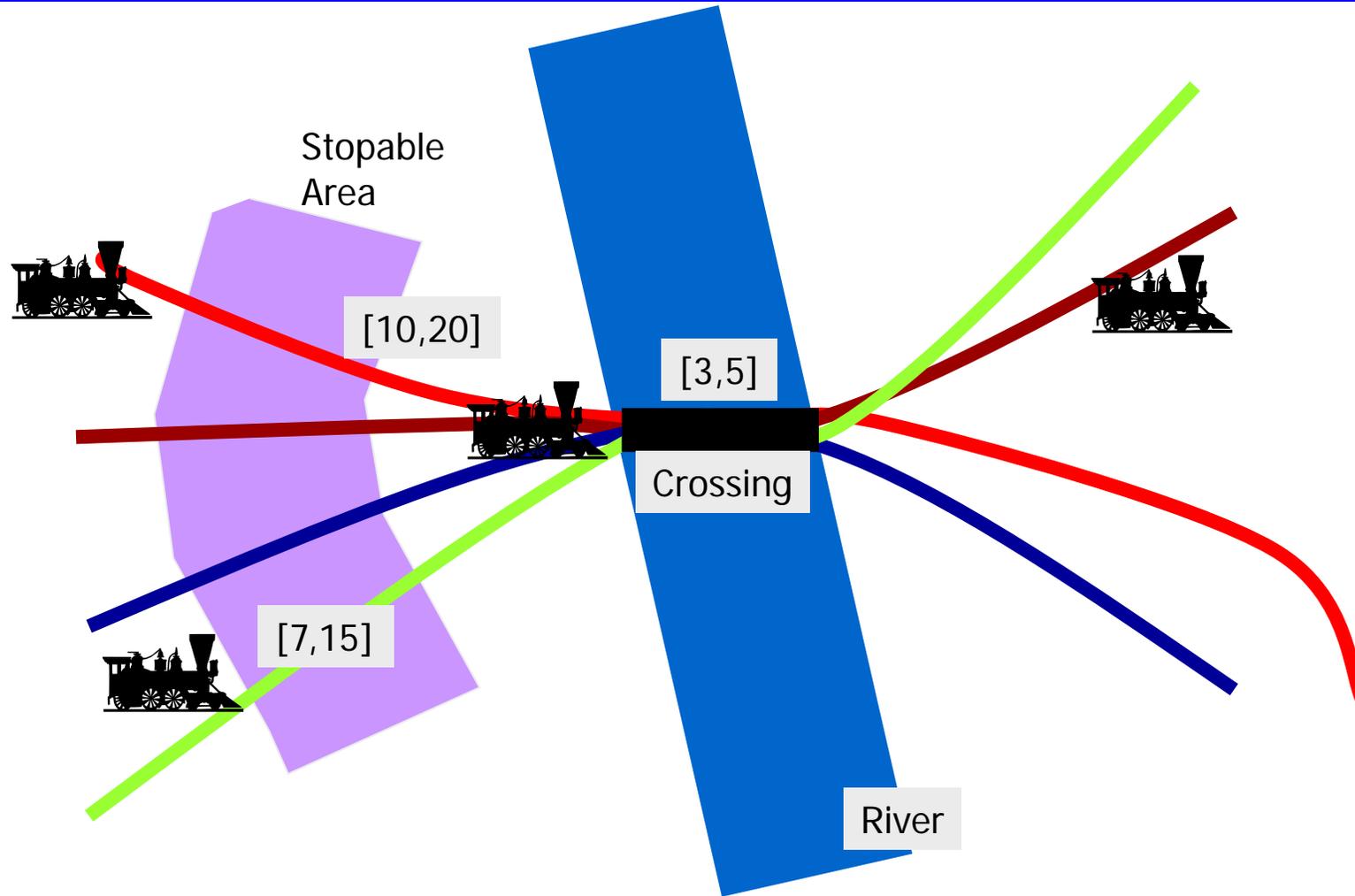
Validierung und Test

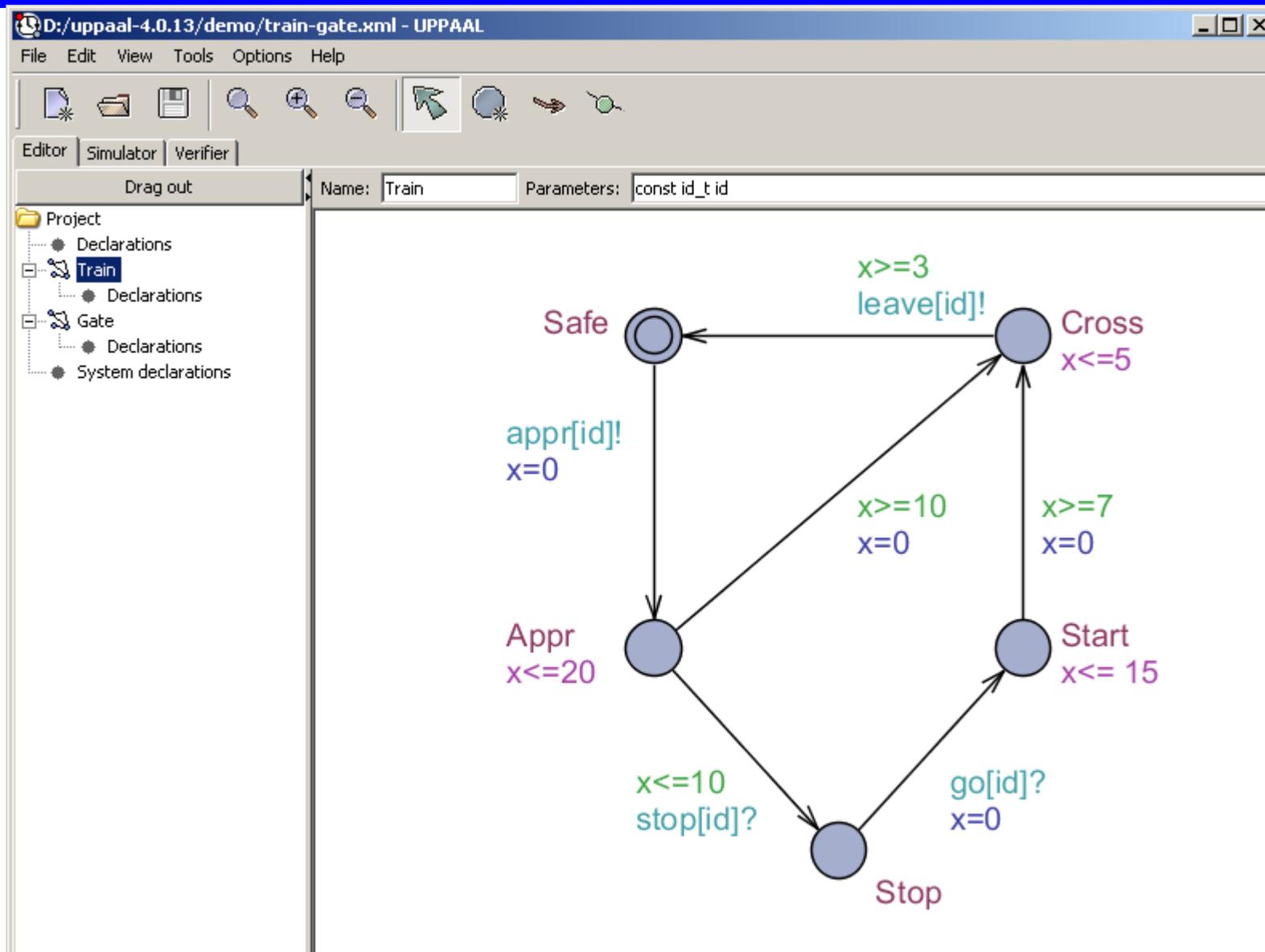
- Testplanung
- Blackbox- und Whiteboxtest
- Modultest
- Testüberdeckung



Alle Tests
erfolgreich!

Beispiel Formale Methoden: Zugangskontrolle für eine Brücke



Modellierung und Automatische Verifikation (mit Uppaal)²⁵

Zusammenfassung

- Software-Produktqualität umfasst die Aspekte
Funktionalität, Zuverlässigkeit, Benutzbarkeit,
Effizienz, Änderbarkeit, Übertragbarkeit
- Software Engineering bedeutet
die Bereitstellung und systematische Verwendung von Methoden, Verfahren
und Werkzeugen zur Entwicklung, Betrieb und Wartung von Software

- **Feedback**

Stellen Sie **viele Fragen** während der Vorlesung!

Ohne **Rückmeldung** können wir nichts verändern!

Durchführung: Am Besten **konstruktiv & offline**.