

1. Vorlesung

Software Engineering I

Studiengang Wirtschafts-Informatik

herzlich willkommen

Prof. Dr.-Ing. Anna Sabine Hauptmann
hauptman@informatik.htw-dresden.de

Z 349 b

WS 2016 © HTW Dresden - Software Engineering I - Prof. Dr.-Ing. Anna Sabine Hauptmann

1. Vorlesung

Was erwarten Sie?/ Was wollen Sie lernen?

Was möchten Sie nach den Lehrveranstaltungen können?

SW-Systeme planen und neu entwickeln bzw. verändern
programmiersprachenübergreifend d.h. Grenzen zwischen den Programmiersprachen abbauen
im Team zusammen arbeiten, Teambildung
Wie kann man Programmcode sauber und schön und performant gestalten?
SW auf verschiedenen Endgeräten → Was ist zu beachten?
Wie kann man effizient testen z.B. Unit-Test, Fehleranalyse
verstehen, begreifen, Prüfung bestehen

WS 2016 © HTW Dresden - Software Engineering I - Prof. Dr.-Ing. Anna Sabine Hauptmann

1. Vorlesung

Was der Kunde erklärte	Was der Projektleiter verstand	Was der Analytiker beschrieb	Was der Entwerfer plante	Was der Programmierer programmierte
Wie das Projekt dokumentiert wurde	Was installiert wurde	Was dem Kunden in Rechnung gestellt wurde	Wie das SW-System gewartet wurde	Was der Kunde wirklich gebraucht hätte

WS 2016 © HTW Dresden - Software Engineering I - Prof. Dr.-Ing. Anna Sabine Hauptmann

1. Vorlesung

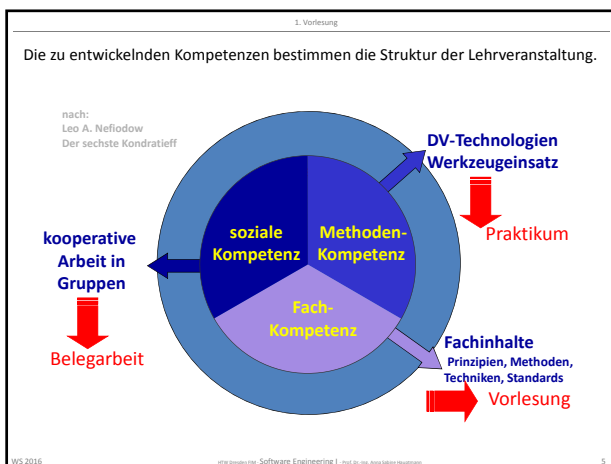
Prof. Dr. Anna Sabine Hauglmann
Professur für Softwareentwicklung

login: SEI2016
PW: Perspektivwechsel

zusätzlich:
Ablage im OPAL
Software Engineering I
HTW Dresden
Fakultät Informatik/Mathematik

Achtung:
kein
Script!

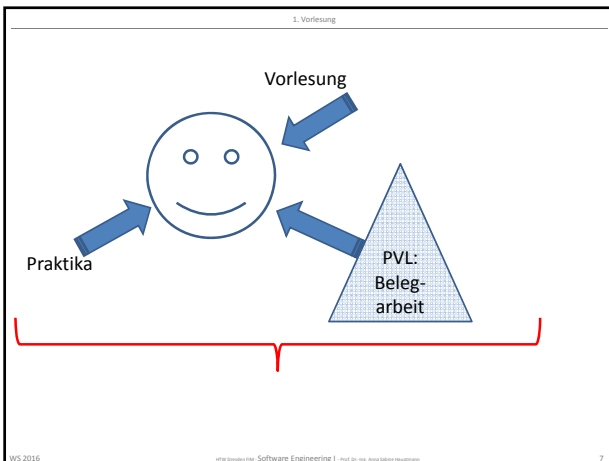
WS 2016

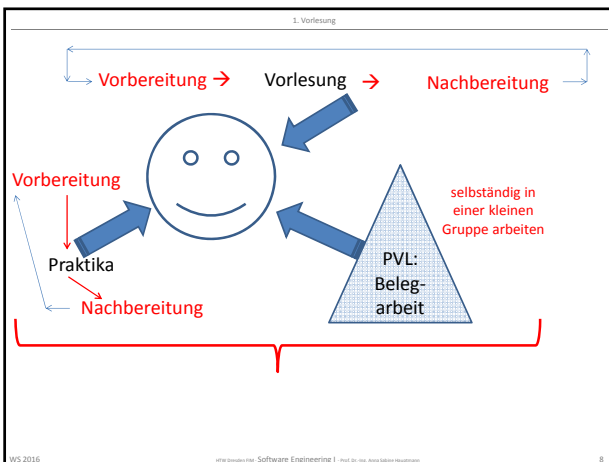


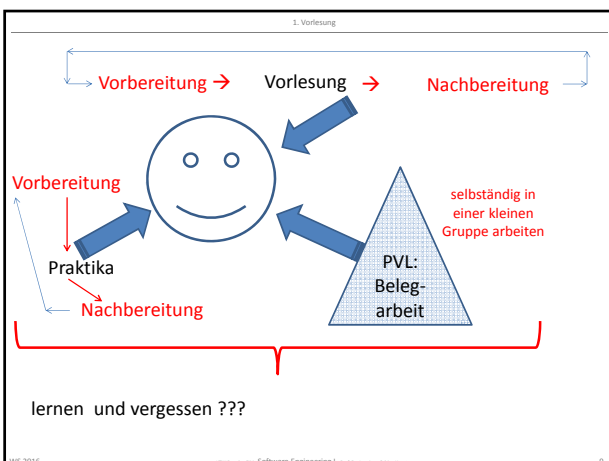
1. Vorlesung

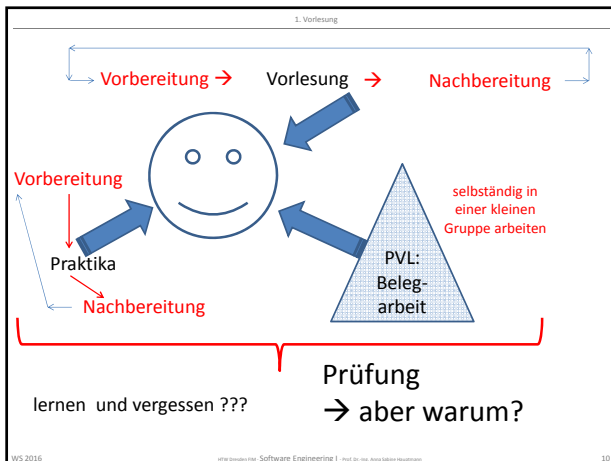
Modul	Software Engineering I
ECTS-Credits	4
Workload	120 h
Präsenzzeit	4 SWS (2/0/2)
Selbststudienzeit	60 h
Prüfungsvorleistung	Belegarbeit
Prüfungsleistung	Schriftliche Prüfung
	120 Minuten ohne Unterlagen

WS 2016









1. Vorlesung

Modul	Software Engineering I
ECTS-Credits	4
Workload	120 h
Präsenzzeit	4 SWS (2/0/2)
Selbststudienzeit	60 h
Prüfungsvorleistung	Belegarbeit
Prüfungsleistung	Schriftliche Prüfung

WS 2016

Software Engineering I

11

1. Vorlesung

Modul	Software Engineering I
ECTS-Credits	4
Workload	120 h
Präsenzzeit	4 SWS (2/0/2)
Selbststudienzeit	60 h
Prüfungsvorleistung	Belegarbeit
Prüfungsleistung	Schriftliche Prüfung

WS 2016

Software Engineering I

12

1. Vorlesung

Modul	Software Engineering I
ECTS-Credits	4
Workload	120 h
Präsenzzeit	4 SWS (2/0/2)
Selbststudienzeit	60 h
Prüfungsvorleistung	Belegarbeit
Prüfungsleistung	Schriftliche Prüfung

Natürlich entscheiden Sie selbst.
Die Lehrveranstaltungen sind eine Einladung an Sie.

Und nun zum Inhalt. Los geht's ...
Aber: Kleine Schritte führen zum Ziel. ➡

WS 2016 Software Engineering I - Prof. Dr. Ingrid Isenhardt 13

2. Vorlesung

Was der Kunde erklärte
Was der Projektleiter verstand
Was der Analytiker beschrieb
Was der Entwerfer plante
Was der Programmierer programmierte

Wie das Projekt dokumentiert wurde
Was installiert wurde
Was dem Kunden in Rechnung gestellt wurde
Wie das SW-System gewartet wurde
Was der Kunde wirklich gebraucht hätte

WS 2016 Software Engineering I - Prof. Dr. Ingrid Isenhardt 14

1. Vorlesung

Wie werden Software-Systeme entwickelt?

Was ist Software-Engineering?

Warum haben Analyse und Definition von Anforderungen an das SW-System große Bedeutung im Entwicklungsprozess?

WS 2016 Software Engineering I - Prof. Dr. Ingrid Isenhardt 15

1. Vorlesung

Ingenieurtechnische Produkt-Herstellung → Die „Bau-Metapher“

Bauherr

Idee: Ein neues Haus

Bau-Architekt

- Reihenhaus, Doppelhaus, ...
- Flachdach, mehrere Etagen
- Anzahl und Nutzung der Räume
- ...

Bauingenieur

- Baumaterialien:
 - Ziegel, Yton, Holz, ...
 - Technologien: Bauteile/Ortbeton
- ...

Bauherr

Maurer, Zimmermann, Klempner, ...

Vertreter der Bauaufsicht

Bauherr

SW Kunde

Was ? Analytiker

Wie ? Entwurfer (Chef-) Architekt

realisieren Programmierer GUI/DB/...

testen Tester

- Projektleiter

- technischer Autor

- Q-Beauftragter

Abstraktion

[illegible][illegible]

1. Vorlesung

Erfolgsquote von Software-Projekten gestern und heute

**Studie des Verteidigungsministeriums der USA
(Ende der 60-iger Jahre)**

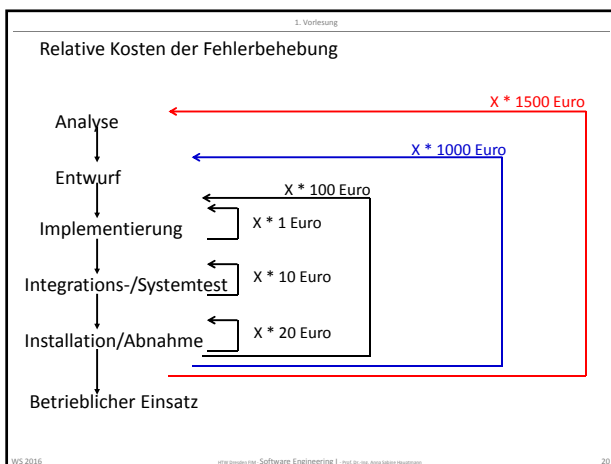
80% werden nie abgeschlossen
15% laufen nur nach aufwendigen, teuren Nachbesserungen
nur 5 % aller SW-Projekte laufen wie geplant

Standish Group, CHAOS Report

1995	2006
30 % aller Software-Projekte scheitern	20 %
53 % werden nicht wie geplant realisiert und/oder die Kunden sind unzufrieden	46 %

60% der Fehler entstehen bereits in der Phase der Analyse

WS 2016 © 2016 Software Engineering I und II der Universität Bayreuth 19



1. Vorlesung

Warum haben Analyse und Definition von Anforderungen an das SW-System große Bedeutung im Entwicklungsprozess?

Auch heute noch werden nur die Hälfte aller SW-Projekte wie geplant realisiert.

60% der Fehler im SW-System resultieren aus Fehlern in der Analysephase, d.h. aus Fehlern bei der Definition der Anforderungen.

Die Behebung von Fehlern aus der Analysephase sind vergleichsweise sehr teuer.

WS 2016 © 2016 Software Engineering I und II der Universität Bayreuth 21

1. Vorlesung	
Einige Definitionen des Begriffes „Software Engineering“ –Teil 1	
Naur, P. Randell, B. 1968 Report on a Conference, Garmisch	The establishment and use of sound engineering principles in order to obtain economically software that is reliable and works efficiently on real machine.
Parnas, D. L. 1974 SW-Engineering or Methodes for the Multi- Person Construction of Multi-Versions Programs Springer Verlag	SW-Engineering ist die Programmierung unter mindestens einer der zwei Bedingungen: - mehr als eine Person schreibt und benutzt das Programm - mehr als eine Fassung des Programmes wird erzeugt
Boehm, B. W. Classics in SW-Engineering 1979, Yourden Press	SW-Engineering ist die praktische Anwendung wissenschaftlicher Erkenntnisse auf den Entwurf und die Konstruktion von Computerprogrammen , verbunden mit der Dokumentation, die zur Entwicklung, Benutzung und Wartung der Programme erforderlich ist.

WS 2016

Software Engineering I und II - Prof. Dr. Ingrid Isenhardt

22

1. Vorlesung	
Einige Definitionen des Begriffes „Software Engineering“ –Teil 2	
Sommerville, I. Software Engineering 1985, Addison Wesley	Software-Engineering befasst sich mit dem Bau von SW-Systemen , die nicht von einem Entwickler allein hergestellt werden können. Software-Engineering beruht auf der Anwendung von ingenieurmäßigen Prinzipien und umfasst sowohl technische als auch nichttechnische Aspekte.
Sommerville, I. Software Engineering 2001, Addison Wesley	Das Software-Engineering ist eine technische Disziplin , die sich mit allen Aspekten der SW-Herstellung beschäftigt.
IEEE Std 610.12 (1990) Standard Glossary of Software Engineering Terminology 1990, IEEE New York	The Application of a systematic, disciplined , quantifiable approach to the development, operation and maintenance of software; that is the application of engineering to software.

WS 2016

Software Engineering I und II - Prof. Dr. Ingrid Isenhardt

23

1. Vorlesung	
Einige Definitionen des Begriffes „Software Engineering“ –Teil 3	
Kahlbrandt, B. Software Engineering 1998, Springer	Software-Engineering ist <ul style="list-style-type: none"> • die Entwicklung, die Pflege und der Einsatz Qualitativ hochwertiger Software unter Einsatz von • wissenschaftlichen Methoden • wirtschaftlichen Prinzipien • geplanten Vorgehensmodellen • Werkzeugen • und quantifizierbaren Zielen
Ein Vorschlag:	Software-Engineering ist <ul style="list-style-type: none"> - die effektive und effiziente Entwicklung und Weiterentwicklung komplexer SW-Systeme - sowie begleitender Dokumente - in einem bewusst arbeitsteilig gestalteten Prozess - unter Anwendung bewährter Prinzipien, Methoden und Modellen.


WS 2016

Software Engineering I und II - Prof. Dr. Ingrid Isenhardt

24

1. Vorlesung

Jochen Ludewig / Horst Lichter „Software Engineering“
Grundlagen, Menschen, Prozesse, Techniken
Auflage 1, dpunkt.verlag 2007 3-89864-268-2, S. 51f

Software Engineering in der Informatik  Hygiene in der Medizin

WS 2016 25

1. Vorlesung

Wie werden Software-Systeme entwickelt?
Bevor mit der Implementierung begonnen wird,
werden die Anforderungen ermittelt und beschrieben.
Danach wird ausgehend von der Beschreibung der Anforderungen eine Struktur festgelegt, nach
der SW-System gebaut wird.
Erst dann beginnt die Implementierung.
Dabei begleiten ständig Tests und Dokumentation die Arbeit.

Was ist Software-Engineering? (siehe weitere Definitionen)

Software-Engineering ist

- die effektive und effiziente Entwicklung und Weiterentwicklung
- komplexer SW-Systeme
- sowie begleitender Dokumente
- in einem bewusst arbeitsteilig gestalteten Prozess
- unter Anwendung bewährter Prinzipien, Methoden und Modellen.

Warum haben Analyse und Definition von Anforderungen an das SW-System große Bedeutung
im Entwicklungsprozess?
Auch heute noch werden nur die Hälfte aller SW-Projekte wie geplant realisiert.
60% der Fehler im SW-System resultieren aus Fehlern in der Analysephase, d.h. aus Fehlern in der
Definition der Anforderungen.

Die Behebung von Fehlern aus der Analysephase sind vergleichsweise sehr teuer.

WS 2016 26

1. Vorlesung

Danke für Ihre Aufmerksamkeit und Ihre Mitarbeit.

WS 2016 27
