

Universität Karlsruhe (TH)

Forschungsuniversität · gegründet 1825

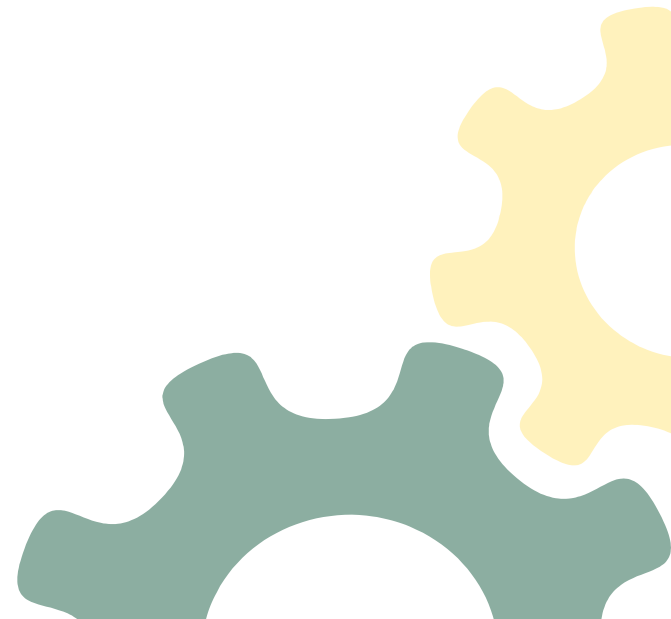
Vorlesung Softwaretechnik I

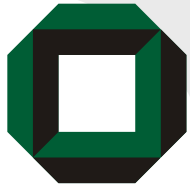
Sommersemester 2009



Fakultät für **Informatik**

Lehrstuhl für Programmiersysteme

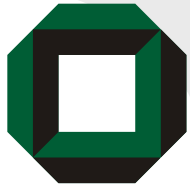




Softwaretechnik (engl. Software Engineering)

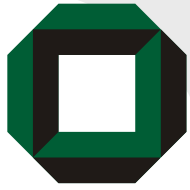
... ist die Lehre von der Softwarekonstruktion.

Diese Vorlesung gibt einen Überblick über die Methoden der Softwarekonstruktion und stellt essentielles Grundwissen für jede/n ingenieurmäßig arbeitende/n Informatiker/in vor.



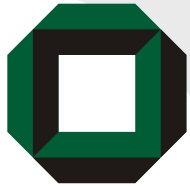
Grobübersicht

- Die Softwaretechnik gibt/sucht Antworten auf die folgenden Fragen:
 - Wie findet man heraus, welche Eigenschaften zu konstruierende Software haben soll (Anforderungsermittlung)?
 - Wie beschreibt man dann diese Eigenschaften (Spezifikation)?
 - Wie strukturiert man die Software so, dass sie sich leicht bauen und flexibel verändern lässt (Entwurf)?
 - Wie verändert man Software, die keine solche Struktur hat oder deren Struktur man nicht (mehr) versteht (Pflege, Restauration, Reengineering)?
 - Wie vermeidet man Mängel in Software oder deckt sie auf (Qualitätssicherung, Test)?
 - Wie organisiert man die Arbeit einer Softwarefirma oder -abteilung, um regelmäßig kostengünstige und hochwertige Resultate termingerecht zu erzielen (Prozessmanagement)?
 - ...und weitere Fragen



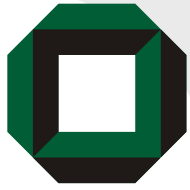
Softwaretechnik >> Programmieren

- Anforderungsermittlung, Spezifikation, Entwurf geschehen vor der eigentlichen Programmierung
- Qualitätssicherung erstreckt sich auf alle Tätigkeiten (sie wird z.B. mittels Durchsichten oder Inspektionen der erstellten Arbeitsprodukte durchgeführt)
- Organisation von kleinen und großen Entwicklerteams geht auch über Programmierung hinaus.



Voraussetzungen für die Vorlesung

- Sie haben gute praktische Kenntnisse der Programmiersprache Java.
- Sie können programmieren und beherrschen mindestens eine Programmiersprache.
- Sie beherrschen die Grundlagen der Objektorientierung.



Vorlesung

Dozent:

Prof. Dr. Walter F. Tichy,

Raum 368, Infobau 50.34

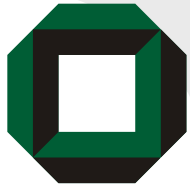
Sprechstunde: freitags 13:00-14:00,

tichy@ipd.uka.de, Tel: 608-3934



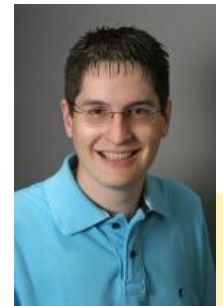
Webseite:

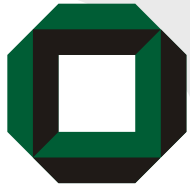
<http://www.ipd.uni-karlsruhe.de/Tichy/>



Organisatorisches

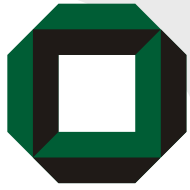
- Übungsleiter:
 - Andreas Höfer, Raum 371, Infobau 50.34
ahoefer@ipd.uni-karlsruhe.de, Tel: 608-7344
 - David Meder, Raum 345, Infobau 50.34
meder@ipd.uka.de, Tel: 608-3972
- Sprechzeiten nach Vereinbarung (≠ jederzeit!)





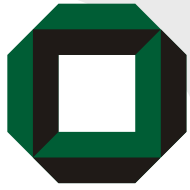
Zum Verhalten in der Vorlesung, Übung, E-Post, etc.

- Wir halten uns an das Prinzip des „vernünftigen Erwachsenen“.
Dieser
 - schätzt die Ressourcen (Zeit und Expertise des Lehrpersonals, techn. Ausrüstung, E-Post-Adressen, E-Post-Verteiler, Webseiten, etc.), die ihm zur Verfügung gestellt werden und nutzt sie zweckentsprechend.
 - erkennt an, dass jeder Fehler machen kann.
 - erhält und pflegt die Lern-Atmosphäre.
 - handelt überlegt.
 - vermeidet Handlungen, die andere beeinträchtigen.
 - sieht sich als Teil einer Gemeinschaft und handelt entsprechend.
 - nimmt nicht nur, sondern gibt auch an die Gemeinschaft zurück.



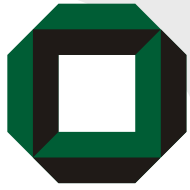
Fragen während der Vorlesung

- Überlegte Fragen sind willkommen.
- Ich bitte sogar darum! Fragen sind eine wichtige Form der Rückkopplung.
- Das Motto „Jede Frage ist wertvoll“ können wir uns bei der großen Zahl von Hörern nicht leisten.
- Daher: bei Unklarheiten sich selbst erst fragen, ob man auch aufgepasst und die Antwort nicht gerade verpasst hat.
- Gerade die Stillen, Zurückhaltenden, Schüchternen und Denker sind aufgefordert, Fragen zu stellen!



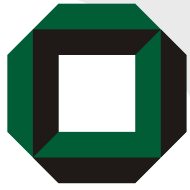
Inhalt

- Was ist Software?
- Warum ist Software so schwer zu entwickeln?
- Warum ist marktreife Software so schwer zu entwickeln?
- Was ist Softwaretechnik?



Was ist Software?

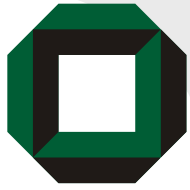
- Software (engl., „Weichware“, Programmatur, im Gegensatz zur Apparatur), Abkürzung: „SW“
- Sammelbezeichnung für Programme und Daten, die für den Betrieb von Rechensystemen zur Verfügung stehen, einschl. der zugehörigen Dokumentation (Brockhaus Enzyklopädie)
- die zum Betrieb einer Datenverarbeitungsanlage erforderlichen nichtapparativen Funktionsbestandteile (Fremdwörter-Duden).



Was ist Software?

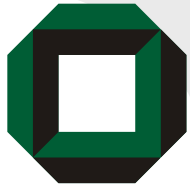
- *Software: Computer programs, procedures, rules, and possibly associated documentation and data pertaining to the operation of a computer system*

(IEEE Standard Glossary of Software Engineering Terminology /ANSI 83/).



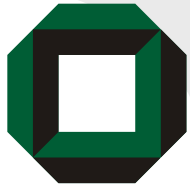
Was ist Software?

- Beispiele für Programme: Quellprogramme, Zwischencode, Objektcode, Bibliotheken, Rahmenarchitekturen, Installationsprogramme, Testprogramme, lauffähige Beispiele.
- Beispiele für zugehörige Daten: Initialisierungsdaten, Hilfsinformation, Fehler und Dialogtexte (evtl. mehrsprachig), Testdaten, erwartete Testausgaben.
- Beispiele für Dokumentation: Anforderungsdokumentation, Architekturbeschreibung, Funktionsbeschreibungen der einzelnen Klassen/Methoden, Test- und Abnahmeprotokolle, Anwendungsbeispiele, Benutzerhandbücher, Fragen und Antworten.
- Mit Software nicht gemeint sind reine Daten, z.B. Musikstücke, Bücher, statische Webseiten, Inhalt von Datenbanken.



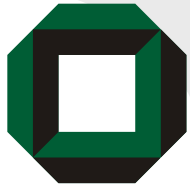
Was ist Software?

- Software-Produkt
 - „Ein Produkt ist ein in sich abgeschlossenes, für einen Auftraggeber oder einen anonymen Markt bestimmtes Ergebnis eines erfolgreich durchgeführten Projekts oder Herstellungsprozesses.“
 - Als Teilprodukt bezeichnen wir einen abgeschlossenen Teil eines Produkts
 - SW-Produkt: Produkt, das aus Software besteht.



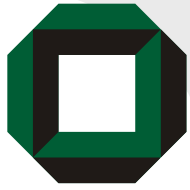
Was ist Software?

- System
 - „Ausschnitt aus der realen oder gedanklichen Welt“
 - bestehend aus Gegenständen (z.B. Menschen, Materialien, Maschinen oder anderen Produkten) und Beziehungen darunter, oder
 - bestehend aus Konzepten und darauf vorhandenen Strukturen (z.B. deren Aufbau aus Teileinheiten oder Beziehungen untereinander, vgl. die Taxonomie der Lebewesen aus der Biologie)“,
 - oder einer Mischung aus realen Gegenständen und Konzepten, sowie Beziehungen darunter, z.B. das Gesundheitssystem (Beziehungen zw. Versicherungsnehmern, Versicherungsgesellschaften, Ärzten, Verträgen, Konventionen, usw.).



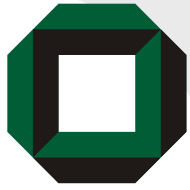
Was ist Software?

- System
 - Systeme sind bei dieser Betrachtungsweise aus Teilen (Systemkomponenten oder Subsystemen) zusammengesetzt, die untereinander in verschiedenen Beziehungen stehen können
 - Systemteile, die nicht weiter zerlegbar sind oder zerlegt werden sollen, werden als Systemelemente bezeichnet



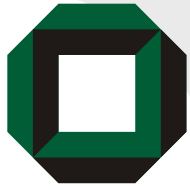
Was ist Software?

- Software-System
 - System, dessen Systemkomponenten und Systemelemente aus Software bestehen



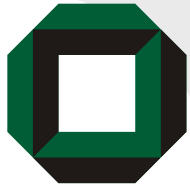
Was ist Software?

- Systemsoftware
 - ... auch Basissoftware genannt
 - Für eine spezielle Hardware oder eine Hardwarefamilie entwickelt, um den Betrieb und die Wartung dieser Hardware zu ermöglichen und zu erleichtern
 - Betriebssystem, Compiler, Datenbanken, Kommunikationsprogramme und spezielle Dienstprogramme (z.B. zur SW-Installation, Benutzerverwaltung, Datensicherung)
 - Orientiert sich grundsätzlich an den Eigenschaften der Hardware, für die sie geschaffen wurde und ergänzt normalerweise die funktionalen Fähigkeiten der Hardware.



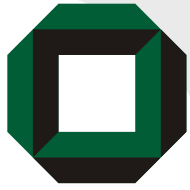
Was ist Software?

- Anwendungssoftware
 - *application software*, Applikationssoftware
 - Löst die Aufgaben des Anwenders mit Hilfe von Rechenanlagen
 - Setzt in Regel auf der Systemsoftware der verwendeten Hardware auf und benutzt sie zur Erfüllung der eigenen Aufgaben
- Computersystem bzw. Datenverarbeitungs(DV)-System
 - Anwendungssoftware +
 - Systemsoftware +
 - Hardware.



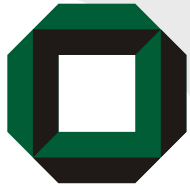
Was ist Software?

- System- vs. Software-Entwicklung
 - Software-Entwicklung
 - Ausschließliche Entwicklung von Software
 - System-Entwicklung
 - Entwicklung eines Systems, das aus Hardware- und Software-Komponenten besteht
 - Bei der Entwicklung müssen zusätzliche Randbedingungen berücksichtigt werden.



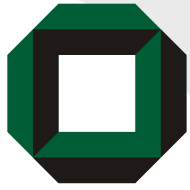
Charakteristiken von Software

- Software ist ein immaterielles Produkt
- Software unterliegt keinem Verschleiß, und es gibt keine Ersatzteile.
- Software wird nicht durch physikalische Gesetze begrenzt
- Software ist **nicht** leichter zu ändern als ein physisches Produkt vergleichbarer Komplexität
- Software ist leichter und schneller zu verteilen und installieren als ein physisches Produkt (bei SW ist nach der Entwicklung keine Produktion erforderlich, sondern nur eine Kopieroperation oder Datenübertragung)
- Software scheint zu altern (Umgebung ändert sich)
- Software ist schwer zu „vermessen“.



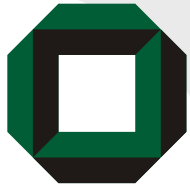
Änderungen in der Software in den letzten Dekaden

- Zunehmende Bedeutung – SW wird zur Wertsteigerungstechnik in vielen Bereichen
- Wachsende Komplexität
- Zunehmende Qualitätsanforderungen
- Nachfragestau und Engpasstechnik
- Zunehmende Standardsoftware
- Zunehmend „Altlasten“
- Zunehmend „Außer-Haus“-Entwicklung.



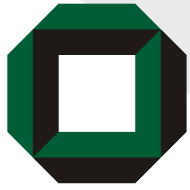
Bedeutung von Software?

- Zunehmende Bedeutung
 - „In zunehmendem Maße entwickelt sich Software zu einem eigenständigen Wirtschaftsgut und spielt eine entscheidende Rolle in der Gesellschaft. Software ist Bestandteil der meisten hochwertigen technischen Produkte und Dienstleitungen geworden. In einigen Bereichen, wie Banken und Versicherungen, werden nahezu alle Dienstleistungen durch Software-Einsatz realisiert“ /BMFT 94, S. 3/



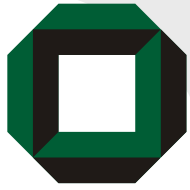
Bedeutung von Software

- Zunehmende Bedeutung (Forts.)
 - Der relative Wertanteil der Software an den Gesamtkosten eines Systems ist in den letzten zwei Jahrzehnten ständig gestiegen.
 - ... dass schon jetzt mehr als die Hälfte der Wertschöpfung von Siemens auf Software-Leistungen entfällt. Diese Entwicklung geht weiter.
Heinrich von Pierer, Siemens AG



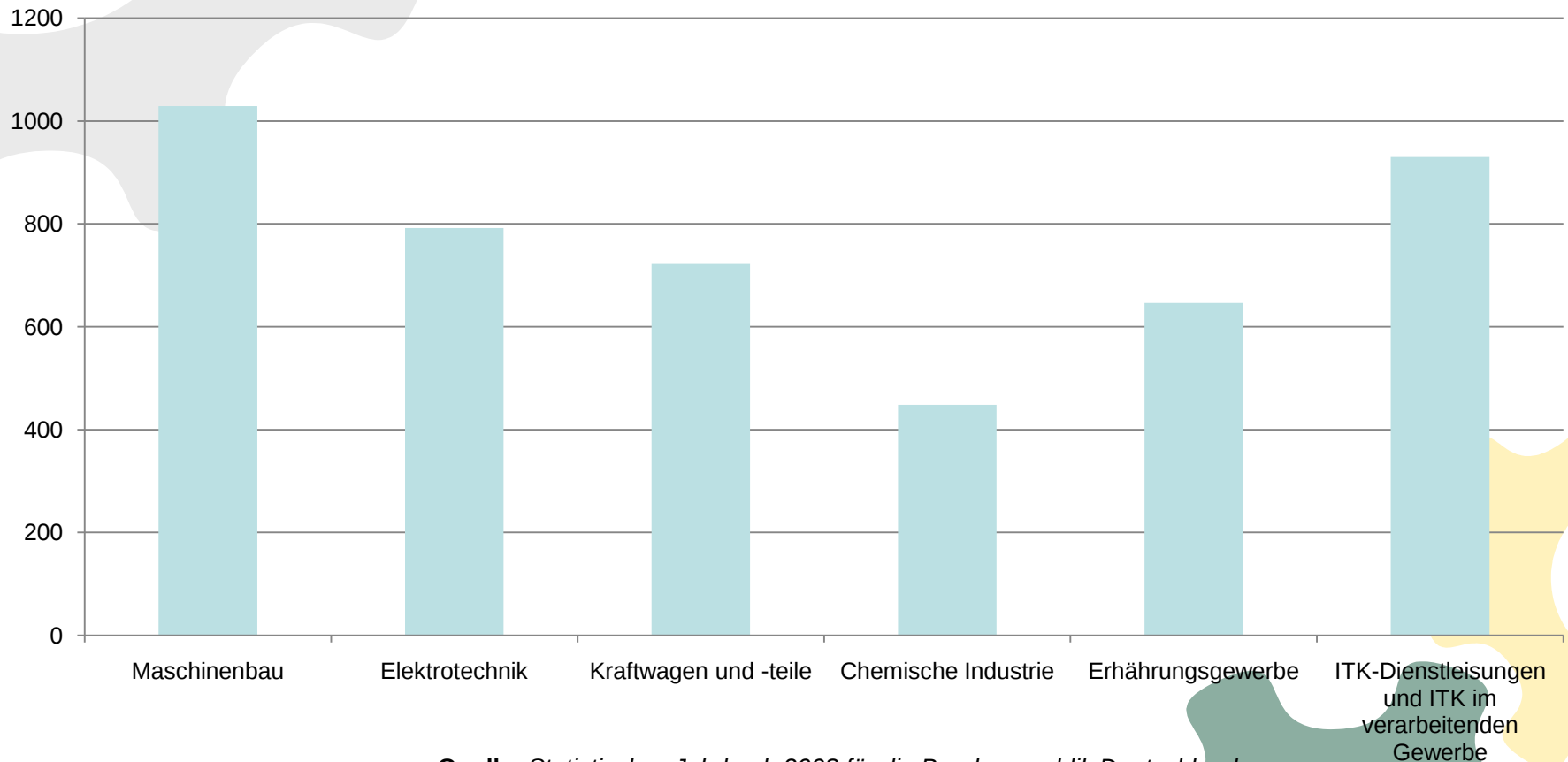
ITK Markt in Deutschland

- ITK: Informations- und Telekommunikationstechnik
- Die ITK-Branche beschäftigte 2008 ca. 828.700 Personen.
- Die ITK erwirtschaftete im Jahr 2008 einen Umsatz 145,5 Mrd. Euro
- davon 45% IT, 45% KT, 10% digitale Konsumelektronik.
- **10% Software**



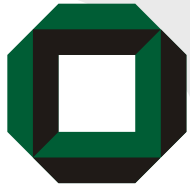
ITK Markt in Deutschland (2007)

Beschäftigte (Tsd.)



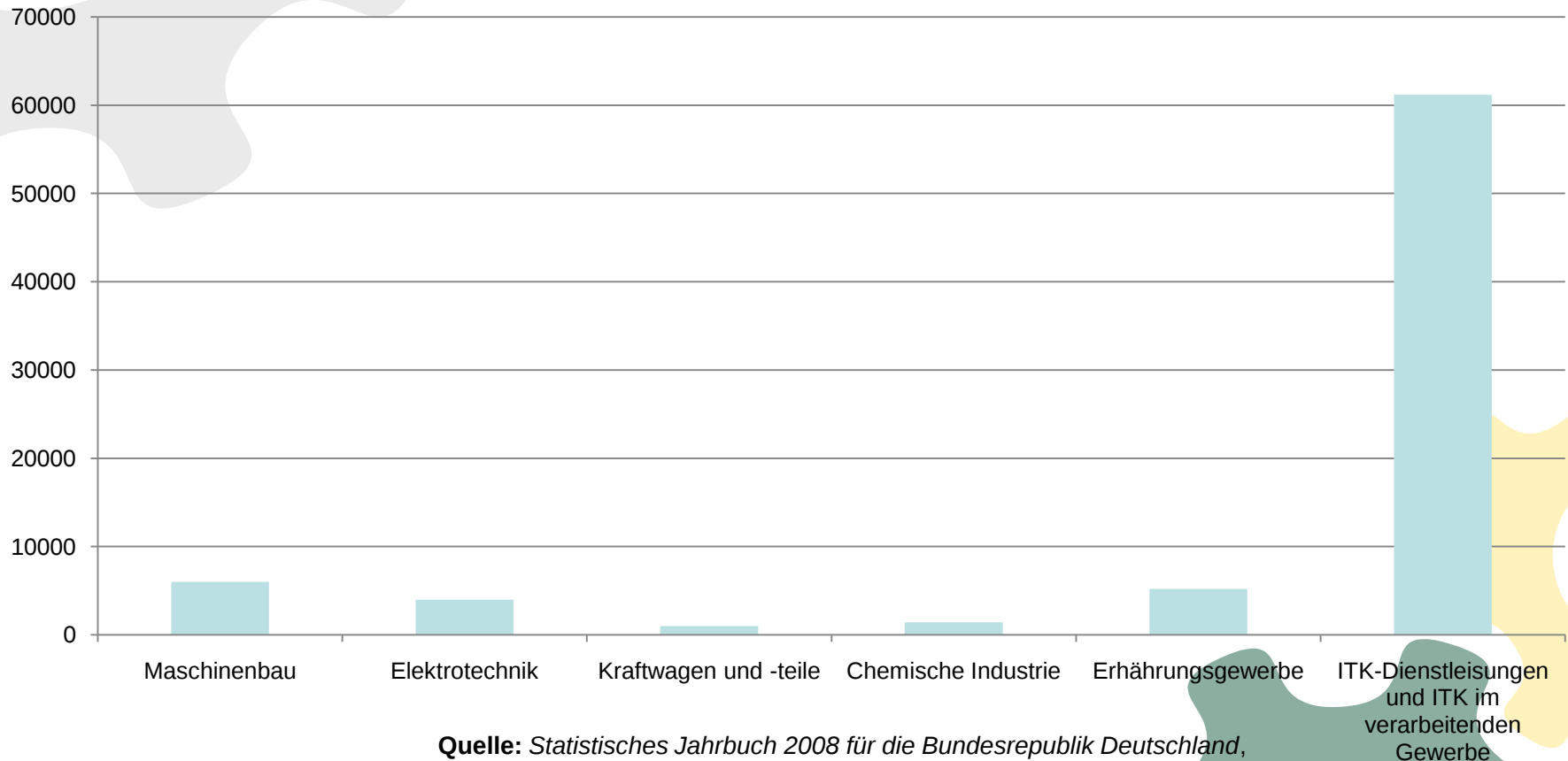
Quelle: Statistisches Jahrbuch 2008 für die Bundesrepublik Deutschland, Statistisches Bundesamt, September 2008.

SWT I - Einführung und Überblick



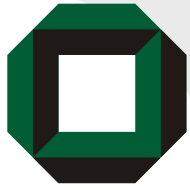
ITK Markt in Deutschland (2006)

Zahl der Unternehmen

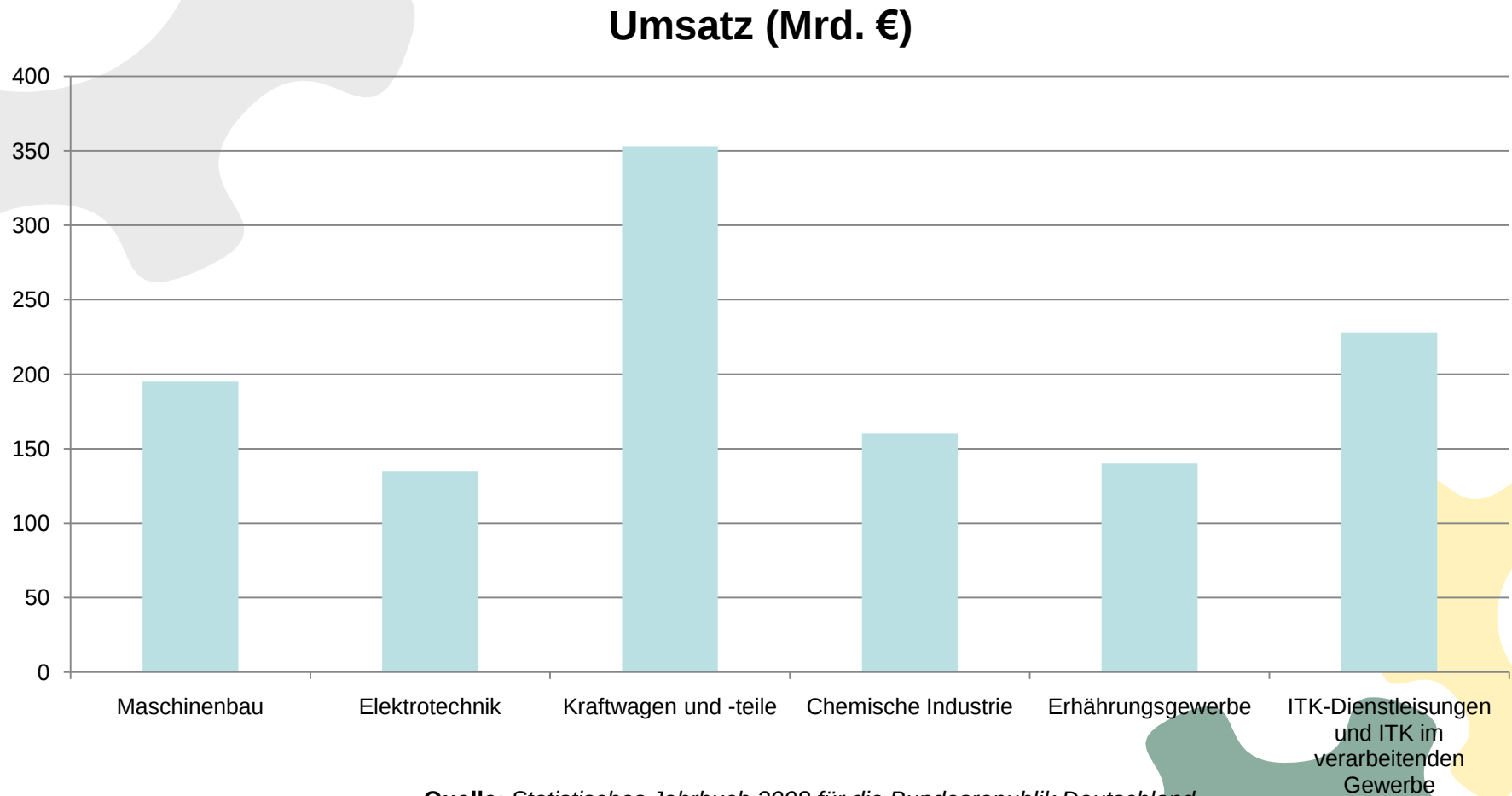


Quelle: Statistisches Jahrbuch 2008 für die Bundesrepublik Deutschland, Statistisches Bundesamt, September 2008.

SWT I - Einführung und Überblick

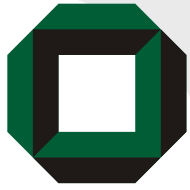


ITK Markt in Deutschland (2006)



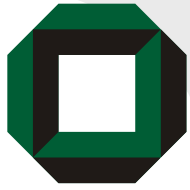
Quelle: Statistisches Jahrbuch 2008 für die Bundesrepublik Deutschland, Statistisches Bundesamt, September 2008.

SWT I - Einführung und Überblick



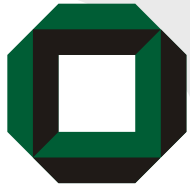
Bedeutung der Software

- Bedeutung der Software-Industrie
 - „In exportorientierten Branchen der deutschen Wirtschaft übersteigt der Software-Anteil an der Wertschöpfung der Produkte häufig die 50%-Marke. Bei Anlagen der digitalen Vermittlungstechnik entfallen bis zu 80% der Entwicklungskosten auf Software“
/BMFT 94, S. 3/.
 - Beispiel: Ein Oberklasse-Fahrzeug enthält 50 bis 70 Mikroprozessoren und drei bis vier Kommunikationsbusse.



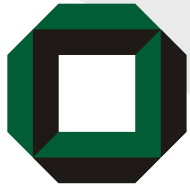
Bedeutung von Software

- Nennen Sie Beispiele, wo Sie von Software in Ihrem täglichen Leben abhängen.



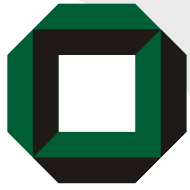
Softwarequalität

- Zunehmende Qualitäts-Anforderungen
 - Für 50% der Ausfälle im industriellen Sektor sind Software-Fehler verantwortlich
 - Nach Cusumano haben sich die gefundenen Defekte in jeweils 1000 Zeilen Quellcode folgendermaßen entwickelt:
 - 1977: durchschnittlich 7- 20 Defekte
 - 1994: durchschnittlich 0,2 - 0,05 Defekte
 - In 17 Jahren konnte die Defektdichte also um ungefähr das 100fache gesenkt werden.



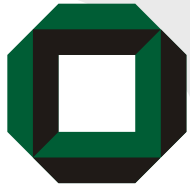
Softwarequalität

- 0,1%-Defektniveau bedeutet:
 - pro Jahr:
 - 20.000 fehlerhafte Medikamente
 - 300 versagende Herzschrittmacher
 - pro Woche:
 - 500 Fehler bei medizinischen Operationen



Softwarequalität

- 0,1%-Defektniveau bedeutet: (Forts.)
 - pro Tag:
 - 16.000 verlorene Briefe in der Post
 - 18 Flugzeugabstürze
 - pro Stunde:
 - 22.000 Schecks nicht korrekt gebucht
- Auch in Zukunft:
 - Massive Qualitätssicherungs-Anstrengungen notwendig.



Softwarequalitätsprobleme

- Beispiel 1: Gepäcktransportsystem des Flughafens Denver
 - Gigantischer Flughafen:
 - doppelte Fläche wie Manhattan,
 - 10 Mal so breit wie Heathrow.
 - 3 Flugzeuge können gleichzeitig landen, in schlechtem Wetter.
 - Gepäcktransportsystem:

34 km	lang
4000	automatische Fahrzeuge
100	Steuerungsrechner
5000	optische Sensoren
400	Funkgeräte
56	Strichcode-Leser

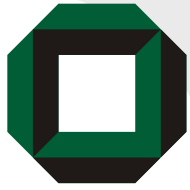


Image U.S. Geological Survey

©2009 Google

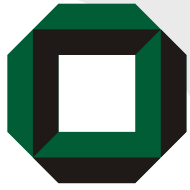
Sichtshöhe 33106 ft





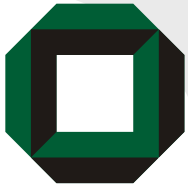
Softwarequalitätsprobleme

- Flughafen Denver
 - Geplante Eröffnung: 31 Okt. 1993
 - Gepäcktransportsystem verzögert wegen Softwareproblemen (sogar Koffer zerfetzt)
 - Erste Verzögerung bis Dez. 1993, dann März 1994.
 - Juni 1994: System noch immer nicht betriebsbereit, keine Schätzung für Zeitpunkt der Inbetriebnahme
 - Inbetriebnahme: Anfang 1995 (16 Monate Verspätung)
- Kosten: 1.1 Million \$ / Tag (Zinsen plus Betrieb)
(siehe Scientific American, Sept. 1994)



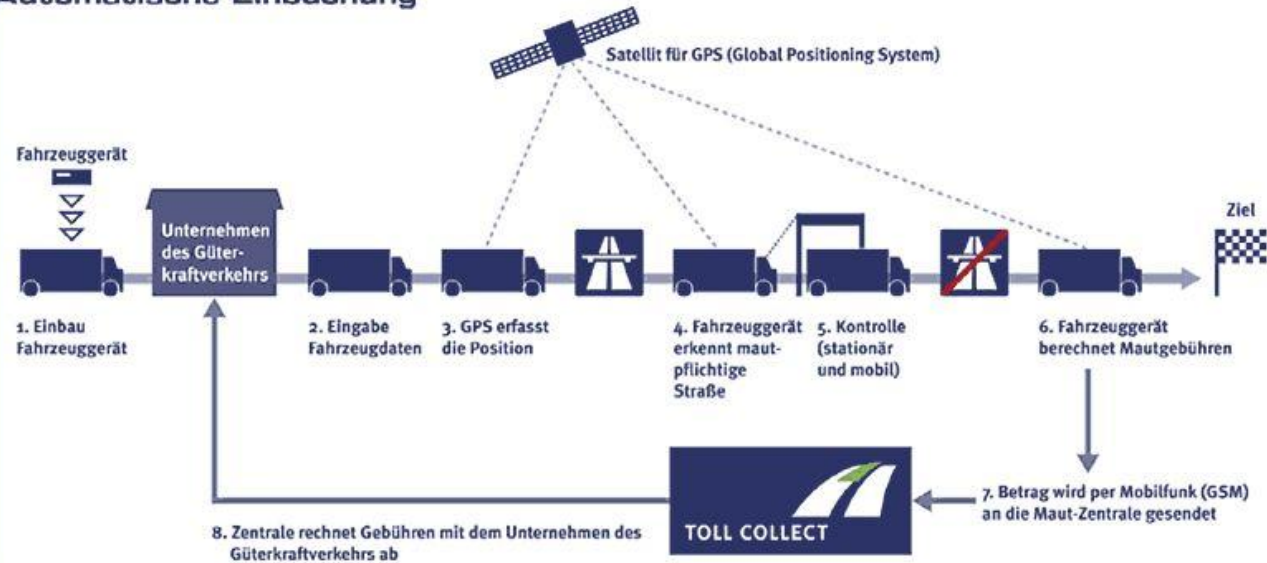
Softwarequalitätsprobleme

- Beispiel 2: Deutschlands TollCollect
 - TollCollect ist ein System zur Mauterhebung für Schwerlasten auf deutschen Autobahnen.
 - Neu: Mauterhebung ohne Anhalten oder Verlangsamung der Lasten; es gibt keine Mautstellen!
 - Statt dessen hat jeder Lasten einen Bordrechner mit GPS-Sensor, der die gefahrenen Strecken erfasst, die fällige Maut berechnet und diese Information periodisch via Mobilfunk (SMS!) an eine zentrale Abrechnungsstelle sendet. Die Abrechnungsstelle erstellt Rechnungen und belastet Kredit- und Flottenkarten.
 - Lasten ohne Bordrechner können ihre Maut an Web-Terminalen an Raststätten vorausbezahlen.
 - Überwachungsbrücken an Autobahnen prüfen korrekte Mautzahlung.



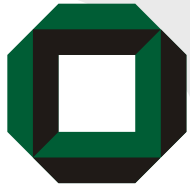
Lkw-Mauterhebung in Deutschland

Automatische Einbuchung



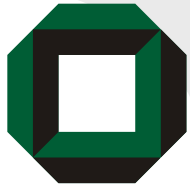
Manuelle Einbuchung im Internet oder am Mautstellen-Terminal





Softwarequalitätsprobleme

- TollCollect:
 - Geplante Inbetriebnahme: 31. August 2003
 - Software Probleme verzögerten den Start (Bordrechner stürzten ab, manche konnten nicht mehr gestartet werden; Zentralsysteme überlastet mit Anfragen nach neuen Autobahnstrecken und aktuellen Tarifen)
 - Alternative: Eurovignette war abgeschafft worden.
 - Reduziertes System ging im Januar 2005 in Betrieb (ohne Aktualisierung von Strecken und Tarifen per Funk)
 - Gesamtsystem ging am 1. Januar 2006 in Betrieb
 - Gesamtverlust: 5600 Million Euro plus Zinsen.



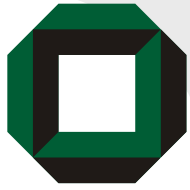
Softwarequalitätsprobleme

Beispiel 3:
Die Ariane 5
Rakete
explodierte 40 s
nach dem Start.
(1996, \$500 Mio.
Schaden).



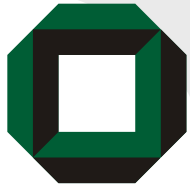
Details: <http://www.around.com/ariane.html>

Video: <http://www.youtube.com/watch?v=kYUrqdUyEpI>



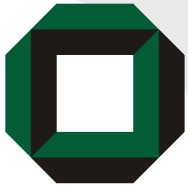
Fallstudie des Absturzes

- Eine Überlauf ereignete sich bei der Konvertierung einer 64-Bit Ganzzahl in eine 16-Bit Ganzzahl in einem Programm genannt „Inertial Reference System“.
- Dieser Überlauf wurde nicht abgefangen, daher hielt die ganze Lagesteuerung an, was die Rakete in eine unzulässige Schräglage brachte, worauf sie sich selbst sprengte.
- Das Programm, das den Überlauf verursachte, war während des Fluges gar nicht benötigt (nur bis -9s Startzeit).
- Es lief aber 50 s lang nach dem Start weiter, um bei einem Abbruch des Starts eine Re-Initialisierung von mehreren Stunden zu vermeiden.
- Das Programm war für die Ariane 4 Rakete entwickelt worden, bei der dieser Überlauf nicht auftreten konnte.
- Der Defekt war also ein Wiederverwendungsfehler, verursacht durch fehlende Spezifikation der Bedingungen, unter denen die Software richtig arbeitete.



Ursache für den Absturz

- 37 Sekunden nach Zündung der Ariane 5 liefert der Sensor für die Horizontale Geschwindigkeit in einer Flughöhe von 3.700 Metern den Wert 32.768,0.
- Es trat ein Überlauf bei der Umwandlung des Wertes 32.768,0 in eine 16-Bit Ganzzahl auf.
- Dieser Fehler wurde jedoch nicht abgefangen.
- Die Software schaltete daher auf den Ersatzrechner um, welcher sich auf Grund des gleichen Problems 72ms zuvor bereits abgeschaltet hatte.
- Die deshalb an den Hauptrechner geschickten Diagnose-Daten wurden als Flugbahndaten interpretiert.
- Daraus resultierten unsinnige Steuerbefehle für die Triebwerke, was zu einer extremen Lage der Rakete führte.
- Da die Ariane 5 auseinanderzubrechen drohte, sprengte sie sich selbst (39 Sekunden nach der Zündung).



Ariane 5 - Navigationssystem

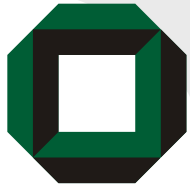
```
...  
declare  
  vertical_veloc_sensor: float;  
  horizontal_veloc_sensor: float;  
  vertical_veloc_bias: integer;  
  horizontal_veloc_bias: integer;  
...  
begin  
  declare  
    pragma suppress(numeric_error, horizontal_veloc_bias);  
  begin  
    sensor_get(vertical_veloc_sensor);  
    sensor_get(horizontal_veloc_sensor);  
    vertical_veloc_bias := integer(vertical_veloc_sensor);  
    horizontal_veloc_bias := integer(horizontal_veloc_sensor);  
  ...  
exception  
  when numeric_error => calculate_vertical_veloc();  
  when others => use_irs1();  
end;  
end irs2;
```

16-Bit Ganzzahl mit
Vorzeichen.

Wert: 32768.0

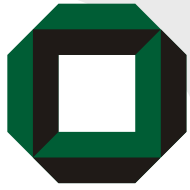
Schalte auf
Ersatzrechner um.

Überlauf bei Umwandlung,
welcher nicht abgefangen
wurde.



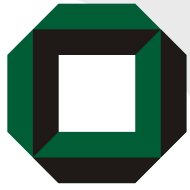
Warum ist Software so schwer zu entwickeln?

- Zunehmend Außer-Haus-Entwicklung
 - Trend:
 - Software nicht selbst entwickeln
 - Sondern: Bei Software-Häusern (auch im Ausland) in Auftrag geben
 - Softwareentwicklung und zugehörige Dienstleistungen werden zunehmend in Niedriglohnländer verlagert.
 - Durch die zunehmende Produktintegration von Software (eingebettete Systeme) wird der Prozentsatz intern erstellter Software nicht drastisch zurückgehen.
 - Typische Geschäftsanwendungen müssen heute in ca. 60 Arbeitstagen erstellt werden.



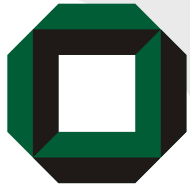
Warum ist Software so schwer zu entwickeln?

- Zunehmend Altlasten
 - Anwendungssoftware wird oft 20 Jahre und länger eingesetzt
 - Da sich die Einsatzumgebung dieser Anwendungssoftware ständig ändert, muss diese Software ebenfalls ständig angepasst werden
 - Diese permanenten Anpassungsprozesse verursachen oft 2/3 aller Software-Kosten.



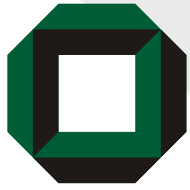
Probleme marktreifer Software

- Anforderungen:
 - Funktionstreue
 - Qualitätstreue
 - Termintreue
 - Kostentreue.



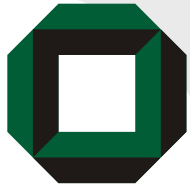
Probleme marktreifer Software

- Spezifische Randbedingungen:
 - Während der Entwicklung ändern sich...
 - die Produktanforderungen
 - neue oder geänderte Funktionen
 - geänderte Benutzungsoberfläche usw.
 - neue Schnittstellen (z.B. Web)



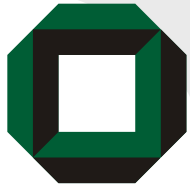
Probleme marktreifer Software

- Spezifische Randbedingungen: (Forts.)
 - Während der Entwicklung ändern sich...
 - die Hardware- und Systemsoftware-Komponenten
 - neuer Prozessor
 - andere Peripherie
 - anderes Fabrikat
 - technische Innovationen.
 - die Software-Methoden und -Werkzeuge, mit denen das Produkt erstellt wird (Produktionsmittel).

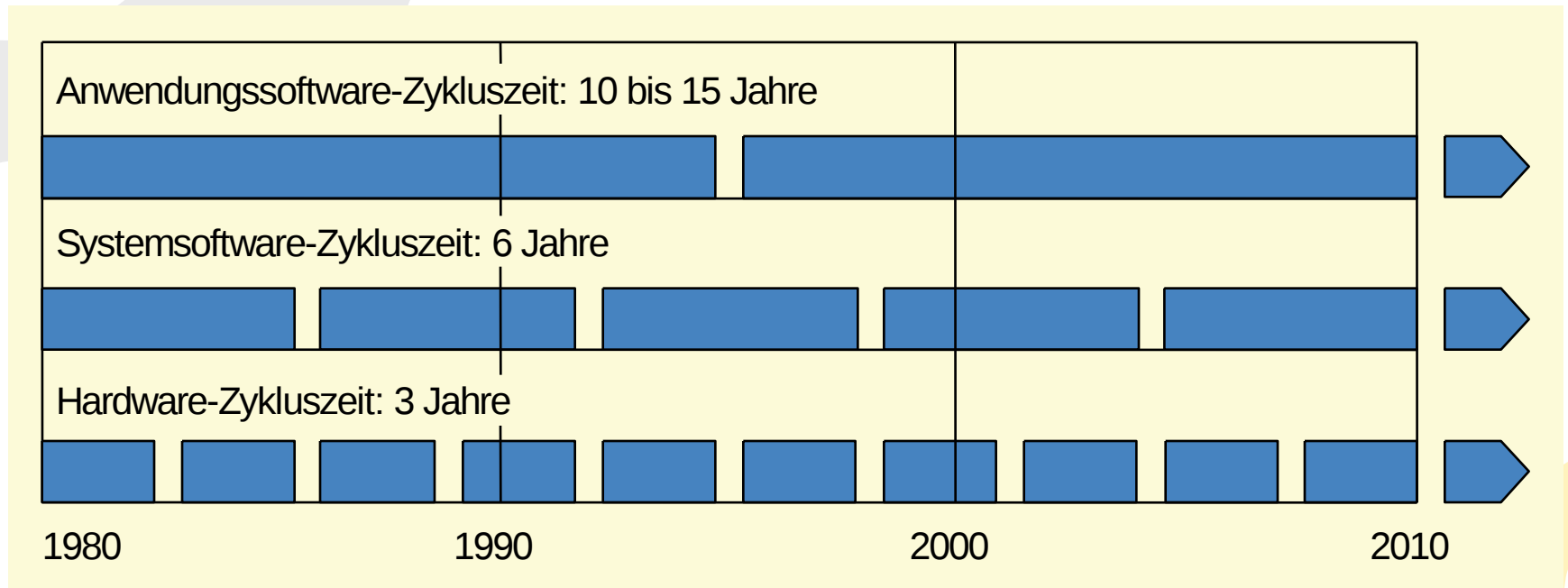


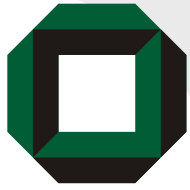
Probleme marktreifer Software

- Spezifische Randbedingungen: (Forts.)
 - Hohe Portabilitätsanforderungen sind einzuhalten...
 - da Software-Produkte eine meist wesentlich längere Lebensdauer besitzen wie die zugrunde liegende Hardware
 - oft auf mehreren Plattformen (HW & Systemsoftware) eingesetzt werden.



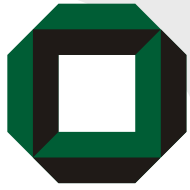
Probleme marktreifer Software





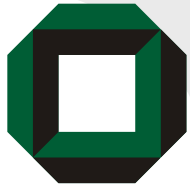
Probleme marktreifer Software

- Entwickelt man heute Anwendungs-SW, dann bedeutet dies, dass...
 - während der Lebenszeit der Anwendungssoftware mindestens einmal die zugrunde liegende System-SW und mindestens zweimal die HW ausgetauscht wird
 - die Zielsysteme, für die Software entwickelt wird, zur Entwicklungszeit oft noch nicht vorhanden sind
 - weltweit vertriebene Software-Produkte länderspezifische Varianten erlauben müssen (muttersprachliche Benutzungsoberflächen).



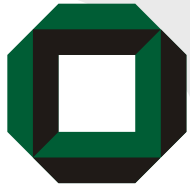
Was ist Softwaretechnik?

- **Softwaretechnik** (engl. Software engineering) ist die technologische und organisatorische Disziplin zur systematischen Entwicklung und Pflege von Softwaresystemen, die spezifizierte funktionale und nicht-funktionale Attribute erfüllen.
- **Softwareforschung** ist die Bereitstellung und Bewertung von Methoden, Verfahren und Werkzeugen für die Softwaretechnik.



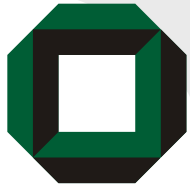
Was ist Softwaretechnik?

- Organisatorische Aspekte:
 - Planung: wie, wo, wann ist was von wem zu tun; was sind die Kosten?
 - Personalfindung, Stellenbesetzung
 - Arbeitsorganisation: Regeln zur Zusammenarbeit (Prozesse, Aufgaben, Zuständigkeiten, Verantwortung, Berichtspflicht)
 - Leitung: Motivation und Kommunikation von Zielen
 - Kontrolle: prüfen, ob Arbeit nach Plan fortschreitet, Anforderung erfüllt werden.



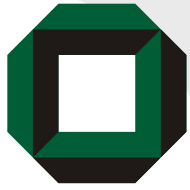
Was ist Softwaretechnik?

- **Systematisch:** es wird nicht herumprobiert oder geraten. Methoden und Werkzeuge werden zielgerichtet und dem Stand der Kunst entsprechend eingesetzt.
- **Entwicklung und Pflege** von Softwaresystemen: Bei erfolgreicher Software ist nach der Entwicklung Pflege erforderlich (Korrekturen, Anpassungen, Erweiterungen). Pflege kann bis zu 2/3 der Gesamtkosten ausmachen.



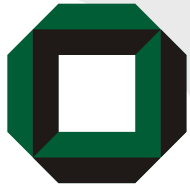
Was ist Softwaretechnik?

- **Funktionale Attribute** spezifizieren die Funktion der Software.
- **Nicht-funktionale Attribute** (auch Qualitätsattribute) spezifizieren, wie gut die Software ihre Funktion erfüllt (wie zuverlässig, wie schnell, wie benutzerfreundlich, wie sicher) aber auch innere Qualitäten wie Änderbarkeit, Dokumentationsgrad.



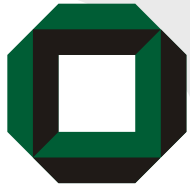
Was ist Softwaretechnik?

- Gesellschaftlicher Auftrag: Software erstellen
 - in ausreichender Menge
 - von zufrieden stellender Qualität
 - termingerecht
 - zu wettbewerbsfähigen Kosten.



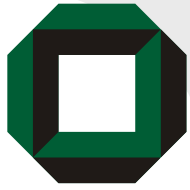
Was ist Softwaretechnik?

- Methoden
 - Planmäßig angewandte, begründete Vorgehensweisen zur Erreichung von festgelegten Zielen
 - Enthalten den Weg zu etwas hin
 - Machen Prinzipien anwendbar
 - Planmäßig: Beim Einsatz einer methodischen Vorgehensweise wird nicht herumprobiert
 - Geben an, welche Konzepte wie und wann verwendet werden, um festgelegte Ziele zu erreichen.
 - Kann durch mehrere Verfahren unterstützt werden



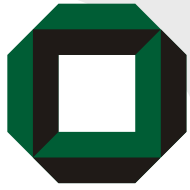
Was ist Softwaretechnik?

- Verfahren
 - sind ausführbare Vorschriften oder Anweisungen zum gezielten Einsatz von Methoden.
 - Beschreiben konkrete Wege zur Lösung bestimmter Probleme oder Problemklassen
 - Ein Verfahren ist im Allg. wegen seiner Beschränkung stärker einsatzbezogen als eine Methode
 - enthalten meist formale Vorschriften und bilden den Inhalt von Standards.



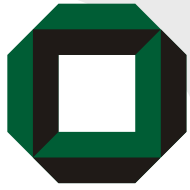
Was ist Softwaretechnik?

- Beispiel: OOA (Objektorientierte Analyse)
- Methoden (Vorgehensweisen):
 - Orientierung an den Daten des Systems (statisches Modell)
 - Orientierung an der Funktionalität des Systems (dynamisches Modell)
- Verfahren:
 - Beschreiben Schritte zum Erstellen des statischen (oder dynamischen) Modells.
 - Bspw.: Klassen, Assoziationen, Attribute, ... identifizieren.



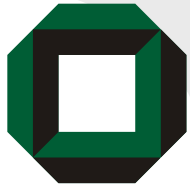
Was ist Softwaretechnik?

- Werkzeuge („tools“)
 - Dienen der automatisierten Unterstützung von Methoden, Verfahren, Konzepten oder bieten Notationen/Sprachen an
- Beispiel
 - Verfahren: Festgelegte Kontrollstrukturen in einer Programmiersprache
 - Werkzeuge: Überprüfung, ob keine *label*-Vereinbarung und kein *goto* verwendet wurde



Was ist Softwaretechnik?

- Werkzeuge (Forts.)
 - Durch Werkzeuge kann die Einhaltung von Methoden, Verfahren, Standards, und Notationen erzwungen und die Produktivität erhöht werden.
 - Werkzeuge automatisieren Tätigkeiten, die...
 - hohe Präzision erfordern
 - oft wiederholt werden müssen
 - eine Überprüfung erfordern

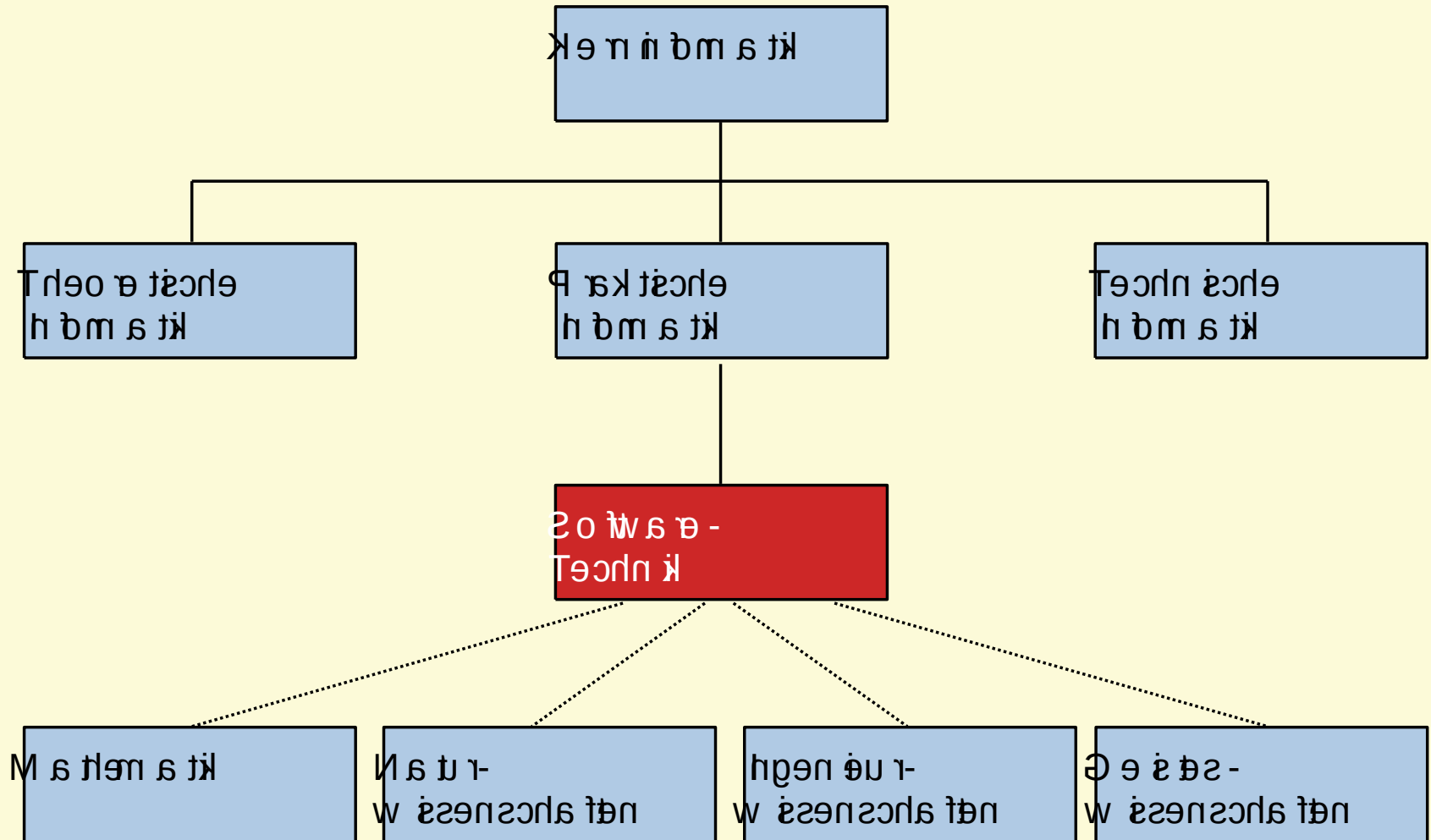


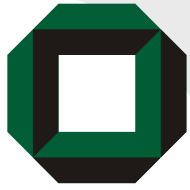
Was ist Softwaretechnik?

- Werkzeuge (Forts.)
 - Um Software zu entwickeln, ist der Einsatz von Werkzeugen notwendig, aber nicht hinreichend
 - CASE
 - *Computer Aided Software Engineering*
 - Software-Entwicklung erfolgt mit Hilfe von Software-Werkzeugen.

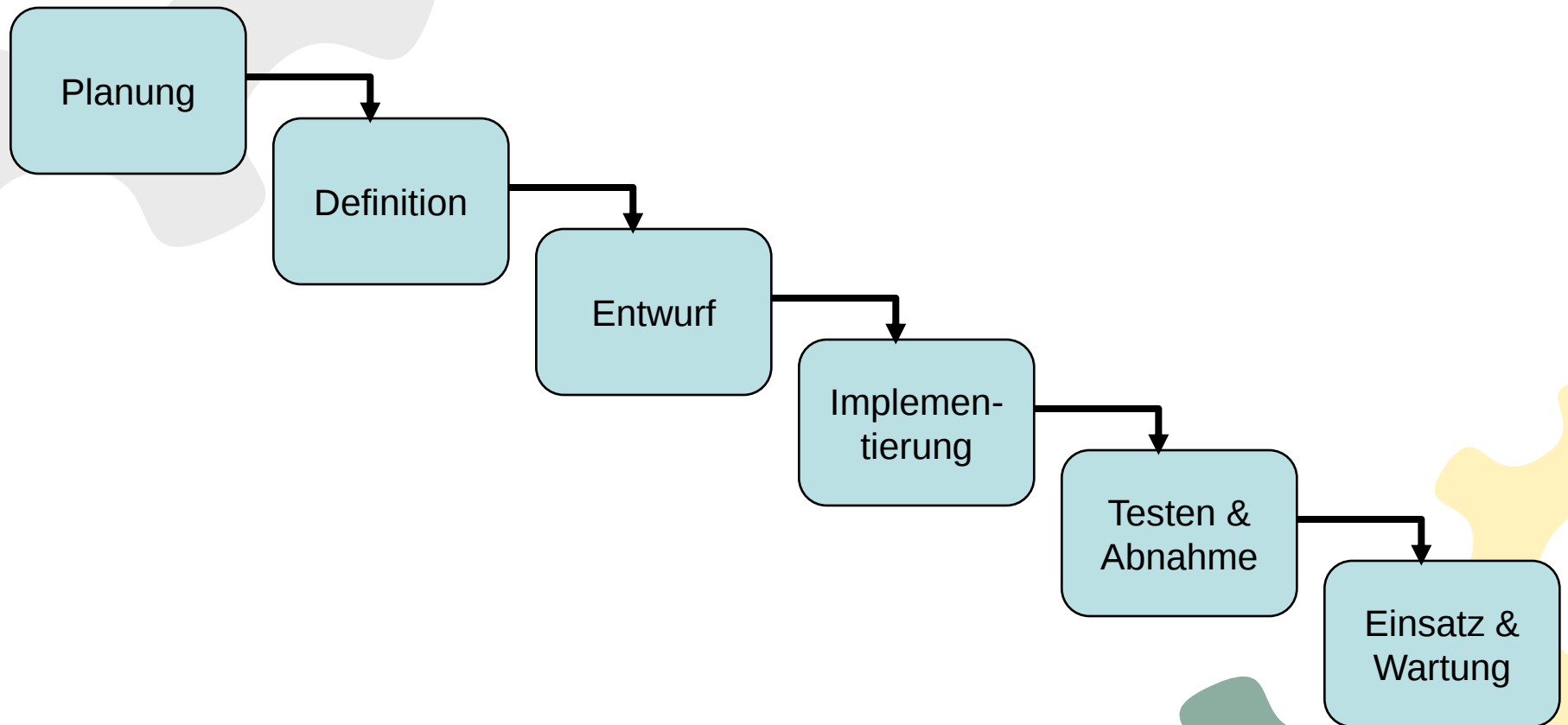


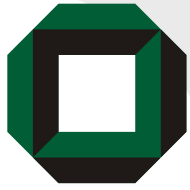
Was ist Softwaretechnik?



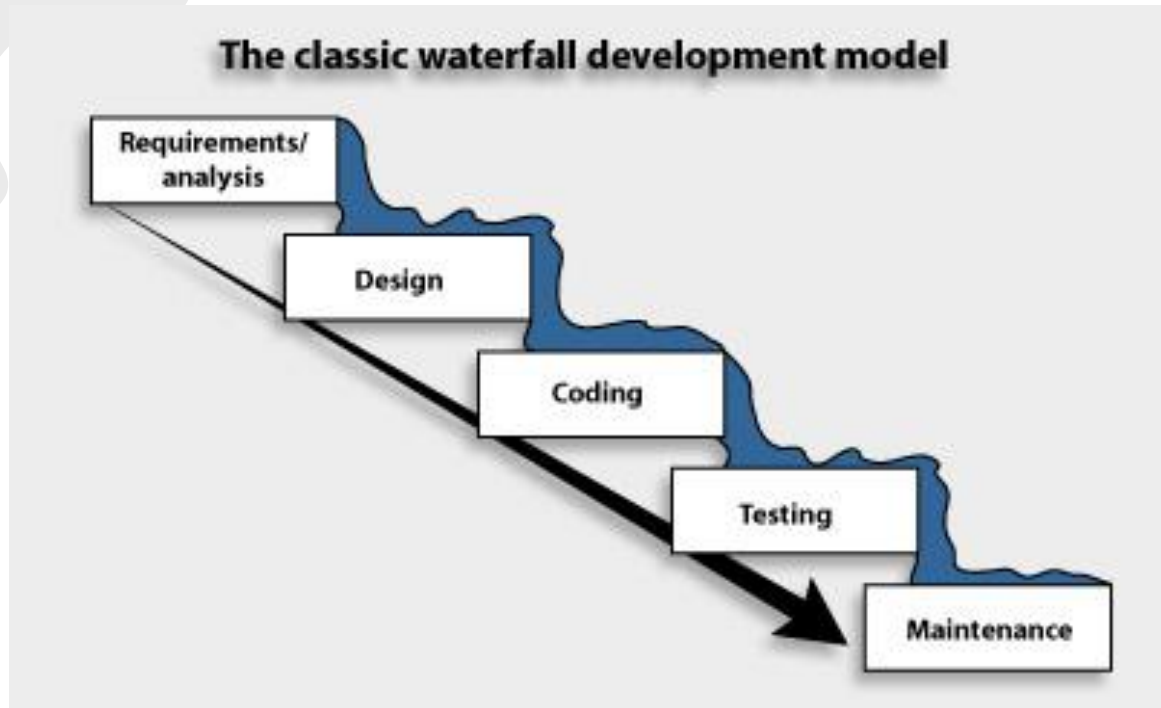


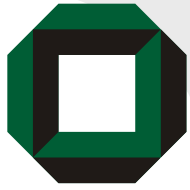
Überblick über den Aufbau der Vorlesung





Die Idee hinter dem Wasserfallmodell





Ohne Kommentar

