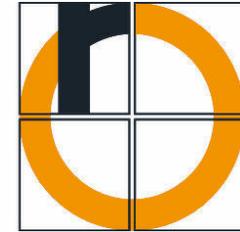




Bavarian Information
and Communication
Technology Cluster

Hochschule Rosenheim
University of Applied Sciences



Studienziel: Software Ingenieur(in), Studentische Projekte mit KMU

Prof. Dr. Gerd Beneken

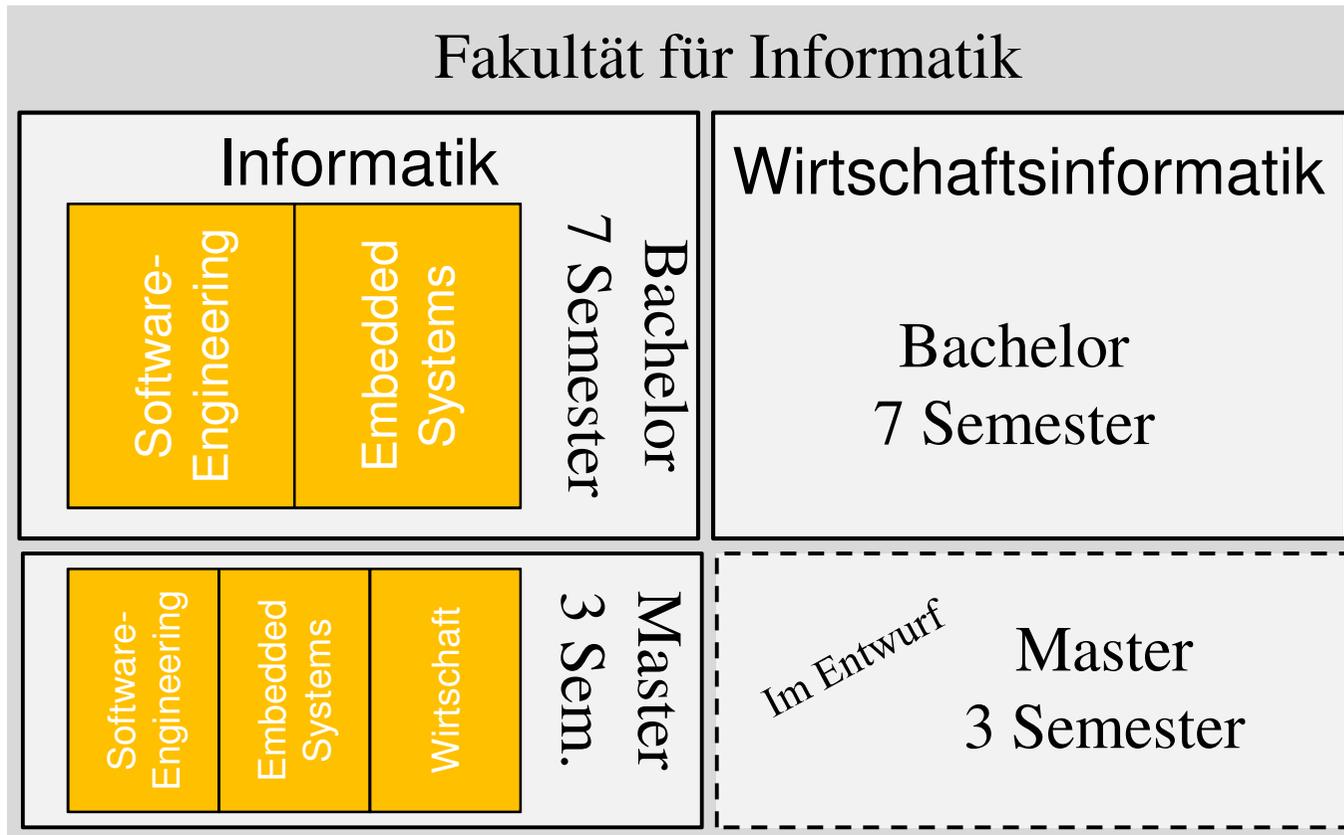
Vortrag am 21.02.2013

Bayerische Lehrstühle & Forschungseinrichtungen öffnen ihre Türen!
BICC Open Labs

Agenda

- Studienziel Software-Ingenieur(in)
- Projekte im Rahmen des Informatik-Studiums
- Projektbeispiele
 - IT Partner in Forschungsprojekten
 - Einzelanfertigungen für genau einen Kunden
 - Unterstützung von Startups / Testen von Geschäftsideen
 - Projekte mit kleinen und mittleren Unternehmen
- Zusammenarbeit mit FH: Nächste Schritte

Informatik an der Hochschule Rosenheim



- ca. 500 Studierende (Bachelor + Master), ca. 10% der Stud.
- ca. 60 - 80 Absolvent(innen) pro Jahr

Studienziel im Bachelor-Informatik Schwerpunkt „Software-Engineering“

Software – Ingenieur(in)

Ab dem ersten Tag produktiv einsetzbar

Mit *solidem Informatik-Fundament* (an der richtigen Stelle)

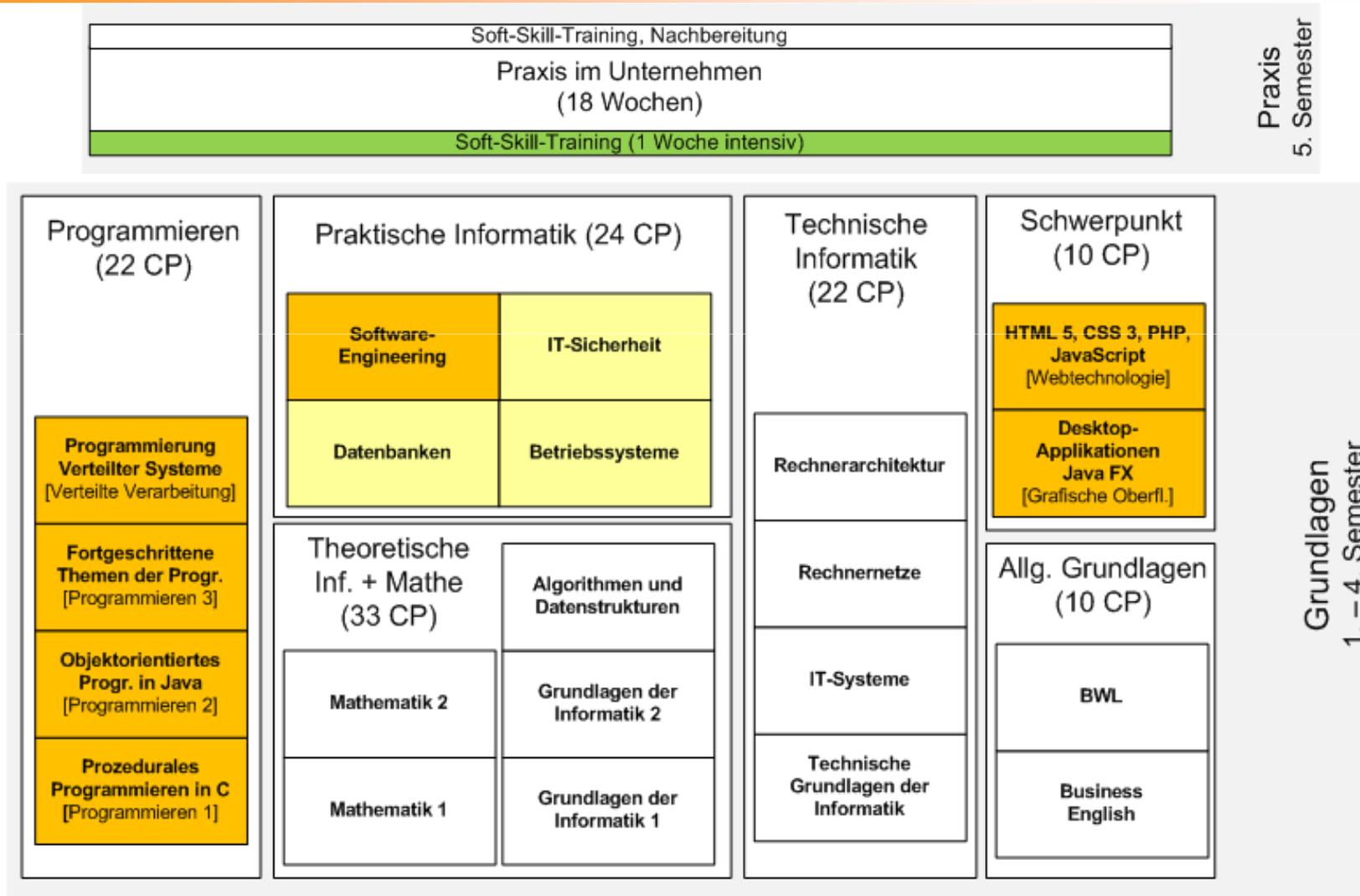
Was macht eine(n) Ingenieur(in) aus?

- **Systematisches, methodisches Vorgehen**
 - Erst Problem genau verstehen, dann Lösung erarbeiten
- **Kostendenken** als Grundlage von Bewertungen
 - Wahl der langfristig kostengünstigsten Alternative
 - Kosten als Vergleichs- und Bewertungskriterium
- **Praktischer Erfolg** als einzige zulässige Beweisführung
 - Erfüllung von Anforderungen muss messbar sein
 - Eine Lösung ist erst fertig, wenn man sie anfassen kann
- **Qualitätsbewusstsein**
 - Ziel unabhängig von Anforderungen: hohe Qualität
- Einführung und Beachtung von **Normen**
- Denken in **Komponenten** / Baugruppen
 - Probleme in kleinere und damit lösbare zerlegen

*) Vgl. auch Ludewig, Lichter: Software-Engineering, dpunkt, 2010

Inhalte des Bachelor-Studiums

Schwerpunkt: Software-Engineering



Schwerpunkt Embedded Systems

(Vertiefungsfächer sind hervorgehoben)

Zusatz Information

Programmieren (22 CP)	Praktische Informatik (24 CP)		Technische Informatik (22 CP)	Schwerpunkt (10 CP)	Grundlagen 1. – 4. Semester	
	Software-Engineering	IT-Sicherheit				Architektur und Programmierung von Microcontrollern [Embedded Systems]
	Datenbanken	Betriebssysteme				Programmieren in Assembler [Maschinennahe Prg.]
	Theoretische Inf. + Mathe (33 CP)					Allg. Grundlagen (10 CP)
Programmierung Verteilter Systeme [Verteilte Verarbeitung]	Algorithmen und Datenstrukturen	Rechnerarchitektur	Rechnetetze	BWL		
Fortgeschrittene Themen der Progr. [Programmieren 3]	Mathematik 2	Grundlagen der Informatik 2	IT-Systeme	Business English		
Objektorientiertes Progr. in Java [Programmieren 2]	Mathematik 1	Grundlagen der Informatik 1	Technische Grundlagen der Informatik			
Prozedurales Programmieren in C [Programmieren 1]						

Programmierer werden / Entwickler werden

- Programmieren wesentliche, **hochqualifizierte Tätigkeit** dort geschieht ein wesentlicher Teil der Wertschöpfung!
- Erste Tätigkeit nach dem Studium ist in der Regel Programmieren (**danach** mit Berufserfahrung: Berater, Requirements Engineer, QM/QS, PM/PL, Architekt, ...)
 - Software-Architekt muss guter Programmierer sein, sonst „zahnloser Tiger“
 - PL/PM muss Schwierigkeiten in Programmierung einschätzen können (auch wenn das eigentliche Programmieren offshore/nearshore passiert)
 - QM muss verstehen, was er qualitäts-sichert, testet, bzw. testen lässt
- Programmieren ist nicht das Beherrschen irgendeiner Syntax sondern **Problemlösen** mithilfe einer Programmiersprache
- Problemlösen/Programmieren **richtig** zu lernen **dauert Jahre** (Ein „Programmierkurs“ ist bestenfalls ein erster Anfang)

Programmieren im Informatik-Studium der Hochschule Rosenheim

- Bis zum Praxissemester (erste 2 Jahre), jedes Semester Programmieren ...
 - 1. und 2. Semester: Grundlagen des Programmierens (C, Java)
 - 3. Semester: Professionelles Programmieren
 - 4. Semester: Programmierung verteilter Systeme (ab Winter 2014)
- Schwerpunkt Software-Engineering
 - 3. Semester: Architektur / Konzepte der **Web-Entwicklung** (PHP, HTML5, CSS3, Java Script)
 - 4. Semester: Architektur / Konzepte der **Desktop-Entwicklung** (am Beispiel Java FX)
- Ziel: Studierende soweit bringen, dass Sie **sicher einen Arbeitsplatz für das Praxissemester finden.**

Software-Engineering ist nicht UML!

Inhalte der Software-Engineering Vorlesungen

- Anforderungsanalyse und –management, Änderungsmanagement
- Spezifikation (**Domänenmodelle, Anw.fälle, GUI-Mockups,...**)
- Entwurf von Architekturen (**Komponenten und Schnittstellen**)
- Softwaretest und Code-Reviews
- **Versionsmanagement** (SVN, GIT, ...), Konfig. Management
- Qualitätssicherung, **Reviews**
(Inspektionen, Stellungnahmen, Walkthroughs)
- **Teamkoordination** über Aufgabenlisten
- Zusätzlich: Vorlesung zu Projektmanagement

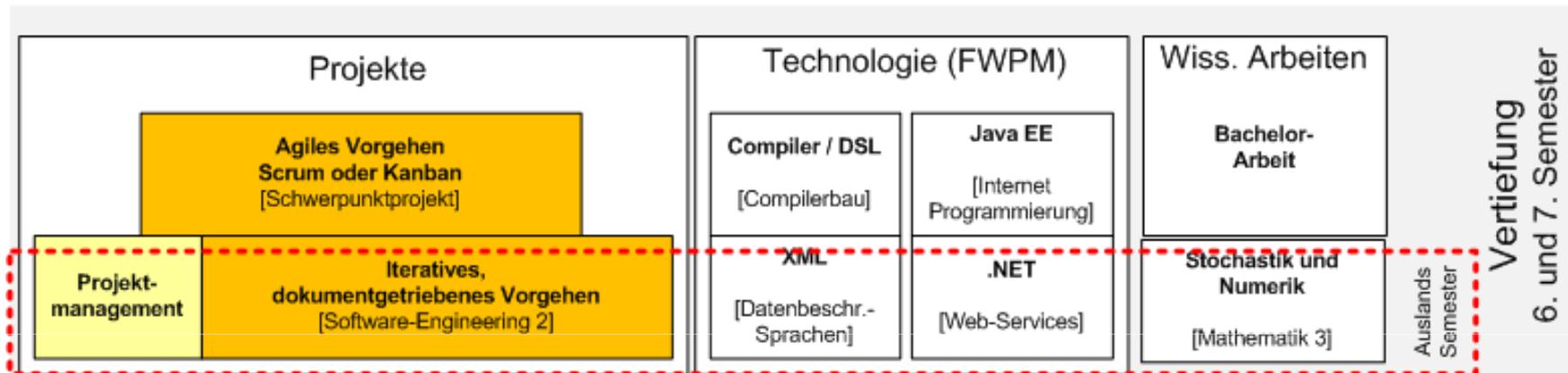
- ***Wichtig: Inhalte werden im realen Projekt ausprobiert***
also nicht: wir modellieren da mal ein Klassendiagramm,
sondern: wir verstehen die reale Domäne besser mit einem Domänenmodell,
das wir z.B. als UML-Klassendiagramm darstellen

Agenda

- Studienziel Software-Ingenieur(in)
- **Projekte im Rahmen des Informatik-Studiums**
- Projektbeispiele
 - IT Partner in Forschungsprojekten
 - Einzelanfertigungen für genau einen Kunden
 - Unterstützung von Startups / Testen von Geschäftsideen
 - Projekte mit kleinen und mittleren Unternehmen
- Zusammenarbeit mit FH: Nächste Schritte

Inhalte des Bachelor-Studiums

Schwerpunkt Software-Engineering: Echte Projekte

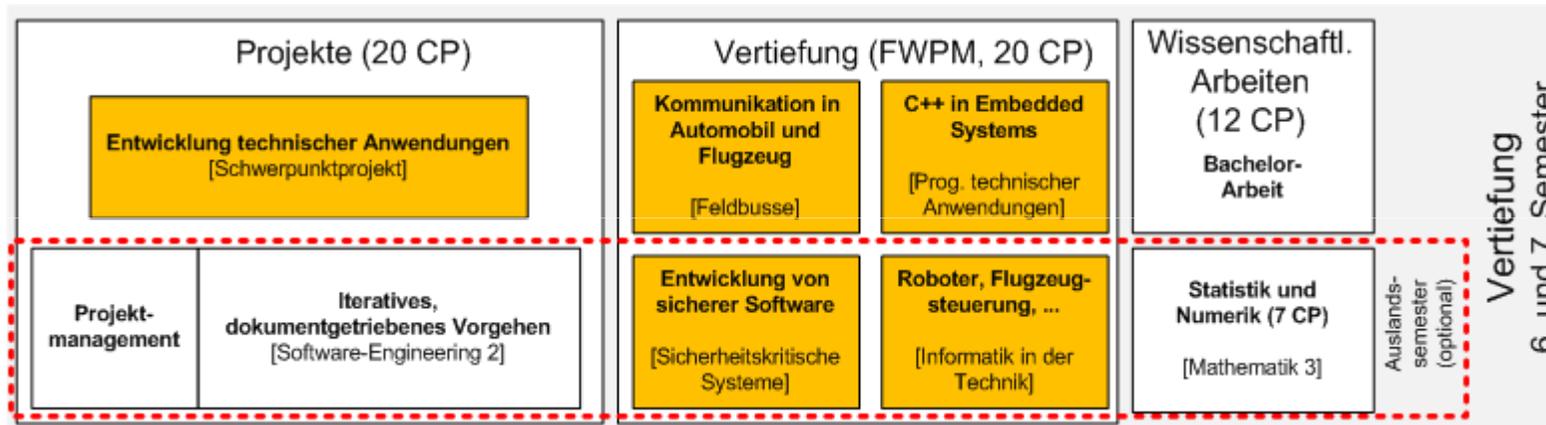


- **2 Pflicht-Projekte** im Rahmen des Studiums, zu je 15 Personentagen
 - „Festpreis-Projekt“ (dokumentgetriebener, iterativer Wasserfall)
 - „Produktentwicklung“ (Scrum)
- Vier Studierende, ein Semester also 60 PT Budget
- **Echte Kunden**, Echte Anforderungen -> Echte Projekte!
- Resultate sollen weiterverwendet werden

Inhalte Embedded Systems

(Vertiefungsfächer sind hervorgehoben)

Zusatz Information



Labor für Software-Technik

Projekttypen

1. Iterativer, dokumentgetriebener Wasserfall (Software-Engineering 2)
 - Dokumentgetriebener Prozess
 - Anforderungen, Spezifikation, Architektur dokumentiert
 - Kurze Implementierungsphase
 - Team wird über den Übungsbetrieb gesteuert
 - **Steuerung wie Festpreis-Projekt** (Änderungsmanagement, Protokolle, ...)
2. Scrum und Agile Methoden (Schwerpunktprojekt)
 - Grundkurs über Agile-Methoden und Scrum
 - Highlight in 2012: Scrum-Training mit Lego bei **msg systems**
 - 3 Sprints mit Lieferung
 - **Steuerung wie in der Produktentwicklung mit Scrum**



Wichtig:

- **Minimal Marketable Featureset** anstreben!
- Dem Kunden genau **zuhören**, realistisch bleiben

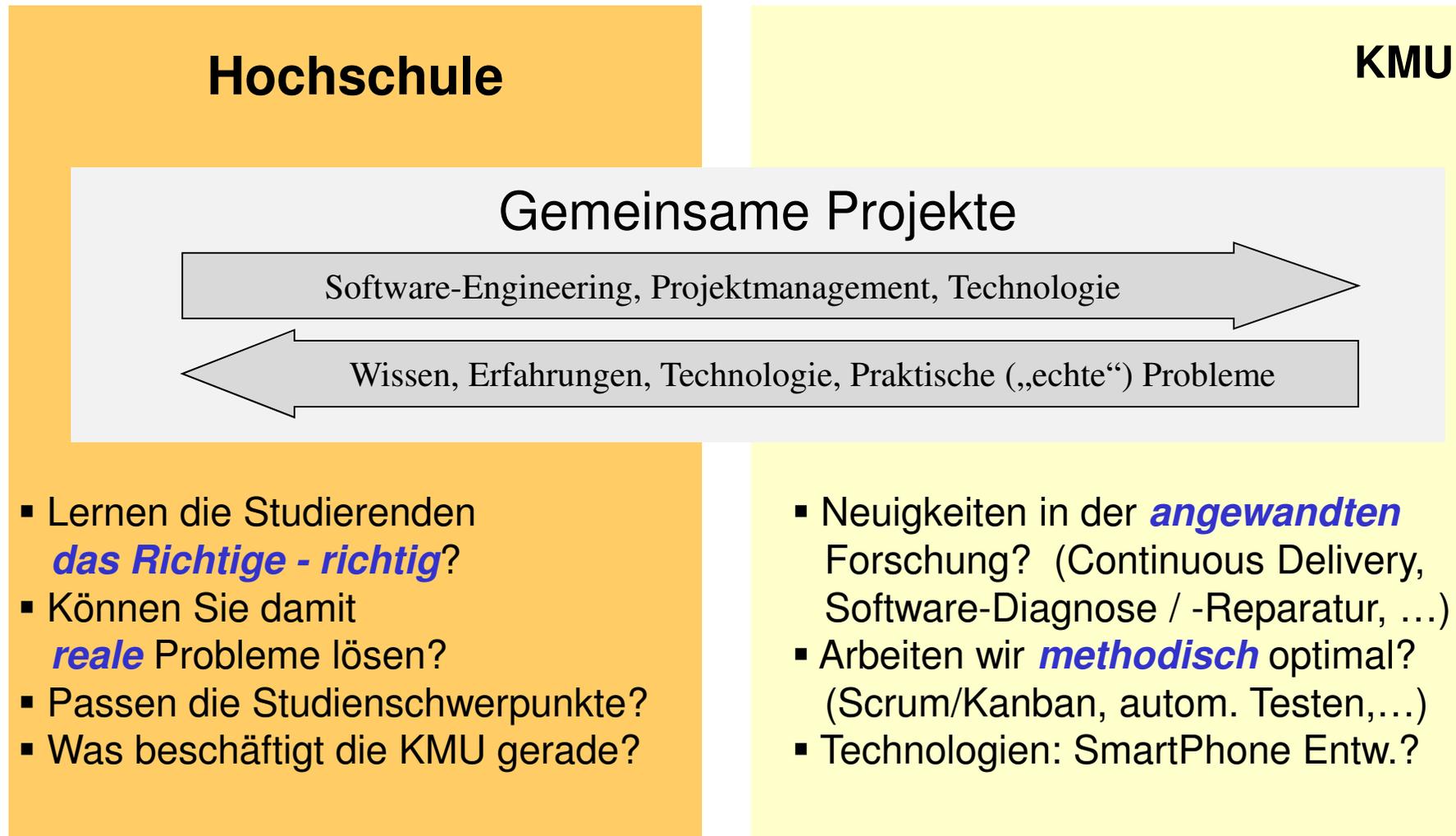
Projekte im Informatik-Studium der Hochschule Rosenheim

- Software-Engineering = *Im interdisziplinären Team kostenbewusst und pragmatisch zu einer passenden (Software-)Lösung gelangen*
- Um das zu verstehen, muss man selber im Team Entwicklungs-Projekte unter Zeit/Kostendruck durchführen
 - Von der Anforderung bis zur Lieferung / Inbetriebnahme
 - Mit echten Kunden sprechen und an diese liefern
- Lernziele
 - Echtes Verständnis des Engineering-Prozesses (agil und dokumentgetrieben)
 - Soft Skills fördern, Raum zum Wachsen geben (Präsentationen, Workshops, Feedback, Konflikte, ...)
 - *Kommunikation* mit einem Kunden (Nicht-Informatiker!)
 - *Arbeiten im Team* (heterogenen, gewisse Konfliktwahrscheinlichkeit)
 - Praktischen Projektalltag vermitteln: Teamkoordination über Aufgabenlisten, Protokolle-Schreiben, Meetings/Workshops organisieren, professionelle Dokumente schreiben, Reviews durchführen, ...)

Rahmenbedingungen für Projekte

- Nutzen für den Projektpartner
 - **Bekannt werden** in der Fakultät als Arbeitgeber (Logo platzieren)
 - Studierende unverbindlich kennenlernen (-> Werkstudenten, ...)
 - Technologie-Wissen „züchten“ (z.B. .NET oder SharePoint)
 - **Prototypen/Showcases erstellen, Wissen aufbauen**
- Aufwand für Partner: ca. 2 Personentage
(für Kickoff, Anforderungsanalyse, weitere Treffen mit Studierenden)
- Projekte: **Unkritische** Prototypen, Showcases, Vorentwicklung, ...
- In der Regel: **Kein Vertrag** zwischen Hochschule und (externem) Projektpartner, d.h. Individuell und mit jedem beteiligten Studierenden zu lösen:
 - Geheimhaltung
 - Reisekosten
 - Lizenzierung der entstandenen Software
(**Software gehört den Studierenden!**)
- ggf. Spende an die Hochschule als Erfolgsprämie

Projekte als Feedbackschleife zwischen KMU und Hochschule

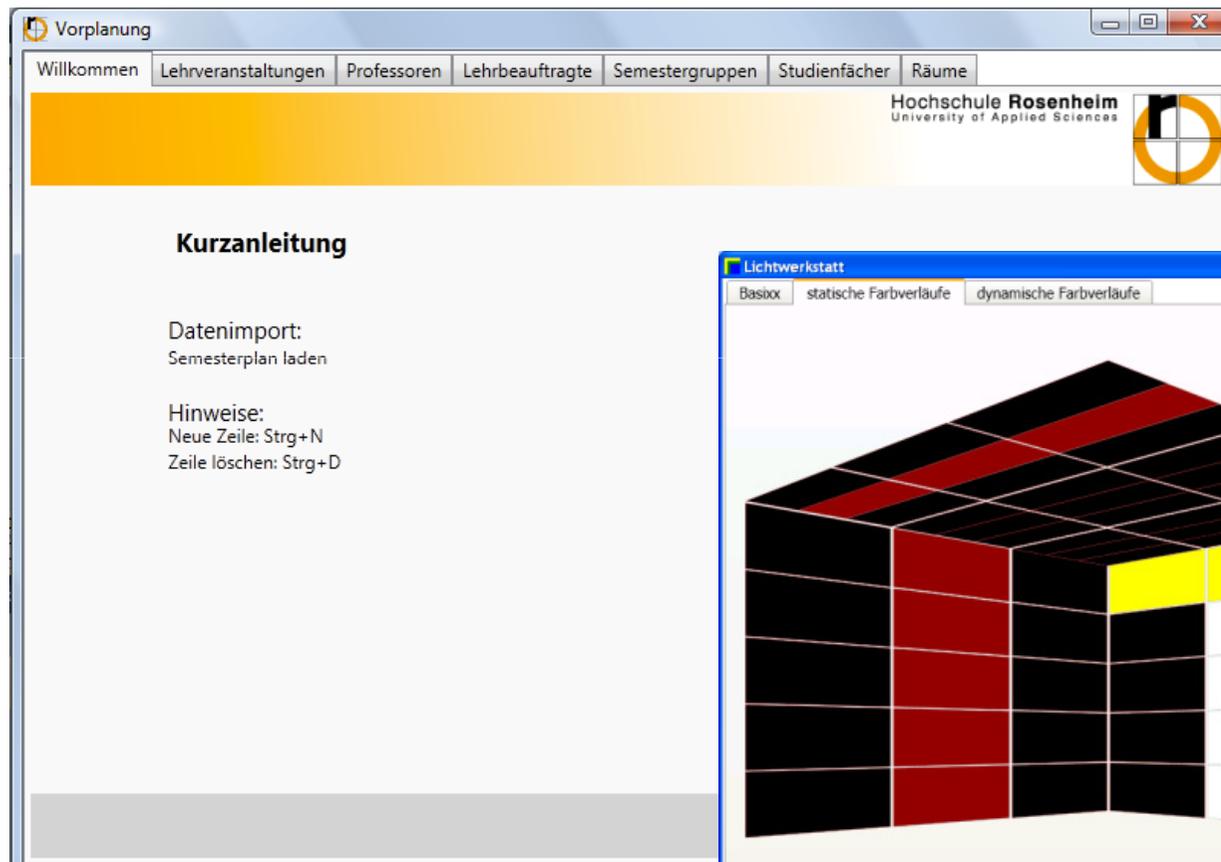


Agenda

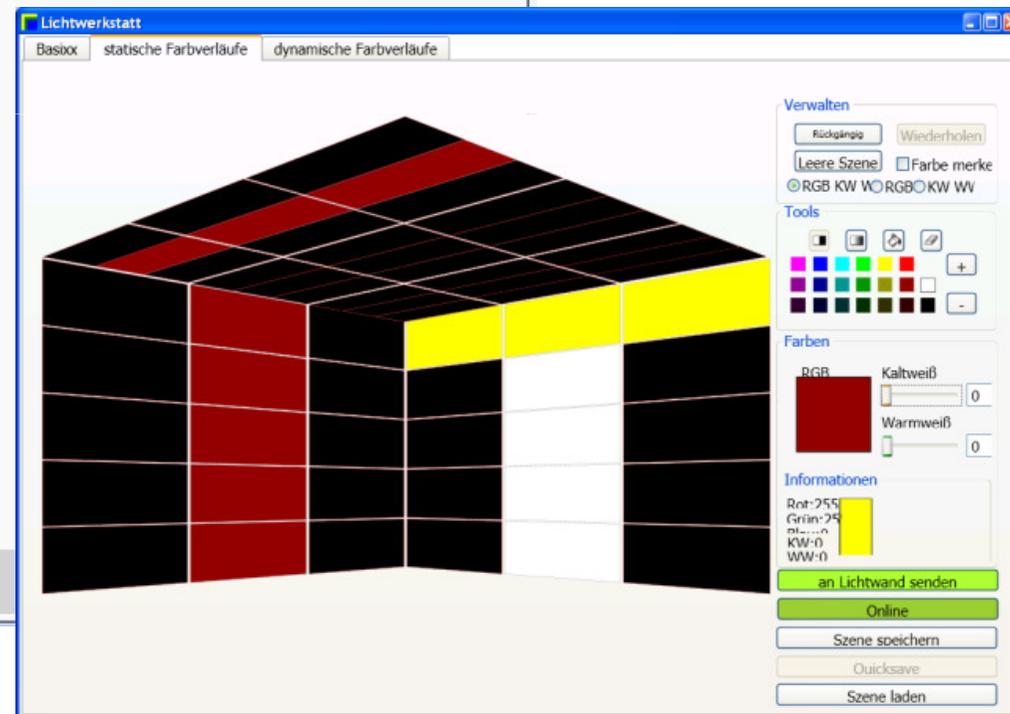
- Studienziel Software-Ingenieur(in)
- Projekte im Rahmen des Informatik-Studiums
- **Projektbeispiele**
 - **IT Partner in Forschungsprojekten**
 - **Einzelanfertigungen für genau einen Kunden**
 - **Unterstützung von Startups / Testen von Geschäftsideen**
 - **Projekte mit kleinen und mittleren Unternehmen**
- Zusammenarbeit mit FH: Nächste Schritte

Interne Anwendungen der Hochschule

Beispiele: Vorplanung und Lichtwerkstatt



Team: Rene Rösner, Andreas Wasner,
Michael Mroz, Manuel Fischhaber, Alfons
Riedhammer, Felix Sauer



Finale Version: Martin Kucich, Team mit 12 Studenten

IT-Partner in Forschungsprojekten

z.B. <http://solar-decathlon.fh-rosenheim.de/>

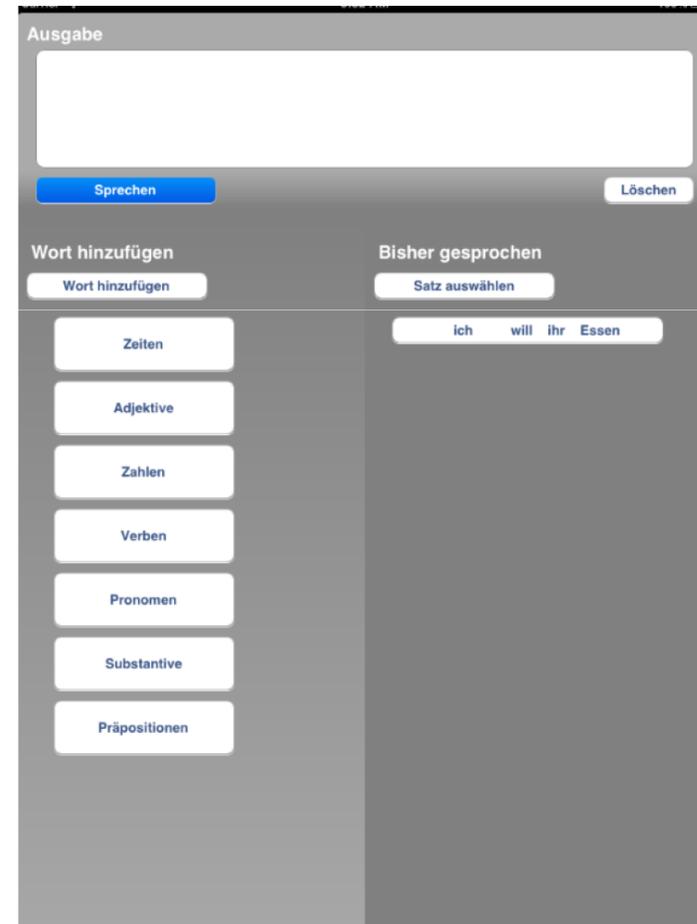
- Gebäudesteuerung des „Solar-Decathlon-Hauses“
 - Wettbewerb sde-europe
 - 2. Platz im internationalen Wettbewerb (Madrid)
- Weiterführung
 - Gestensteuerung des Hauses über Kinect
 - Steuerung über Windows SmartPhone (auch „imagine cup“ Beitrag).



Lichtsteuerung von Anton Koslow + Markus Neuerburg,
Team: Benedikt Gassner und Philip Trojandt

Einzelanfertigungen für genau einen Kunden

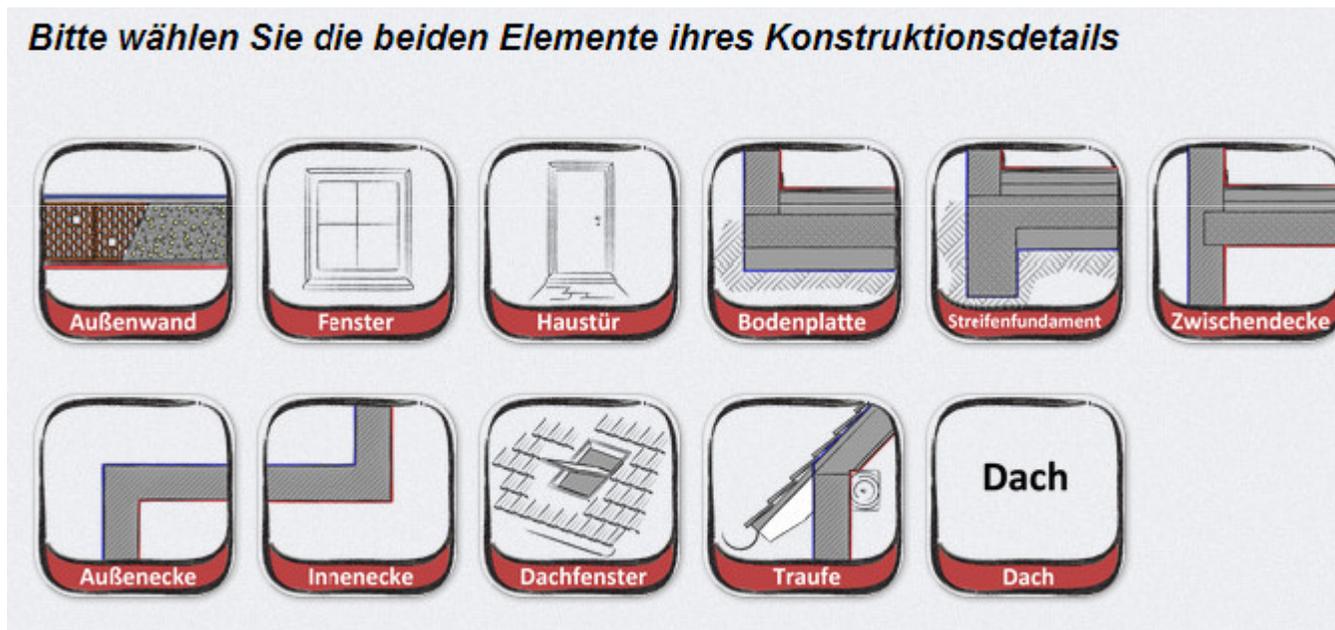
Beispiel: TalkerApp für iPad



Projektteam: Thomas Kreidenhuber, Klaus Voggenauer, Christian Schmid

Unterstützung von Startups / *Geschäftsideen*

Beispiel: Wärmebrückendatenbank



Projektteam: Markus Berwanger, Julia Blum, Sabine Seiler

Projekte mit Unternehmen

Abgeschlossene Projekte

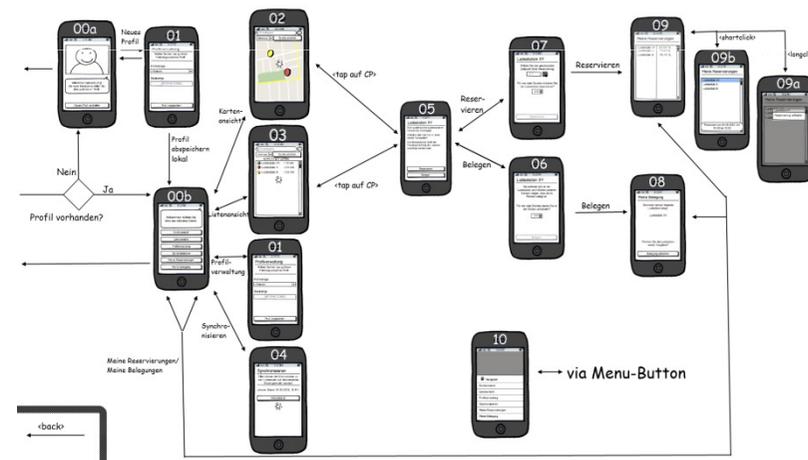
- eridea AG
- Cirquent (jetzt NTT-Data)
- QAware
- iteratec
- Kyocera
- Cassidian/EADS



Beispiel: Therapie-App für Alzheimer-Patienten mit Schön-Kliniken (Rahman Arezowye, Philipp Fahlteich, Benjamin Heller, Wolfgang Herbst)

Geplante Projekte, Sommer 2013

- Halvotec
- Accantum
- iteratec
- ...



Beispiel: Reservierung einer Ladesäule für E-Fahrzeuge
Projekt für die eridea AG (Alexander Zenger, Max Oberberger, Wolfgang Herbst, Obermayer)

Agenda

- Studienziel Software-Ingenieur(in)
- Projekte im Rahmen des Informatik-Studiums
- Projektbeispiele
 - IT Partner in Forschungsprojekten
 - Einzelanfertigungen für genau einen Kunden
 - Unterstützung von Startups / Testen von Geschäftsideen
 - Projekte mit kleinen und mittleren Unternehmen
- **Zusammenarbeit mit FH: Nächste Schritte**

Wenn Sie Interesse an einer Kooperation haben ...
Bitte um Kontaktaufnahme!

Informatik

Prof. Dr.

Gerd Beneken

- **Hochschule für angewandte Wissenschaften
Fachhochschule Rosenheim**
Hochschulstr. 1
83024 Rosenheim
Telefon: 08031 / 805 – 2513
E-Mail: gerd.beneken@fh-rosenheim.de

Hochschule **Rosenheim**
University of Applied Sciences

