



# **Fail Better: Erfahrungen aus 12 Jahren studentischer Projekte**

Prof. Dr. Gerd Beneken

SagWas Workshop am 01.10.2015, Cottbus

Derek Sivers: Why you need to fail

<https://www.youtube.com/watch?v=HhxcFGuKOys>

---

# Agenda

---

- Heimatplanet von G. Beneken
- Warum Projekte mit echten Kunden?
- **„Failure“ Storys**
- Lehrraum ist auch mal Leerraum
- Fazit



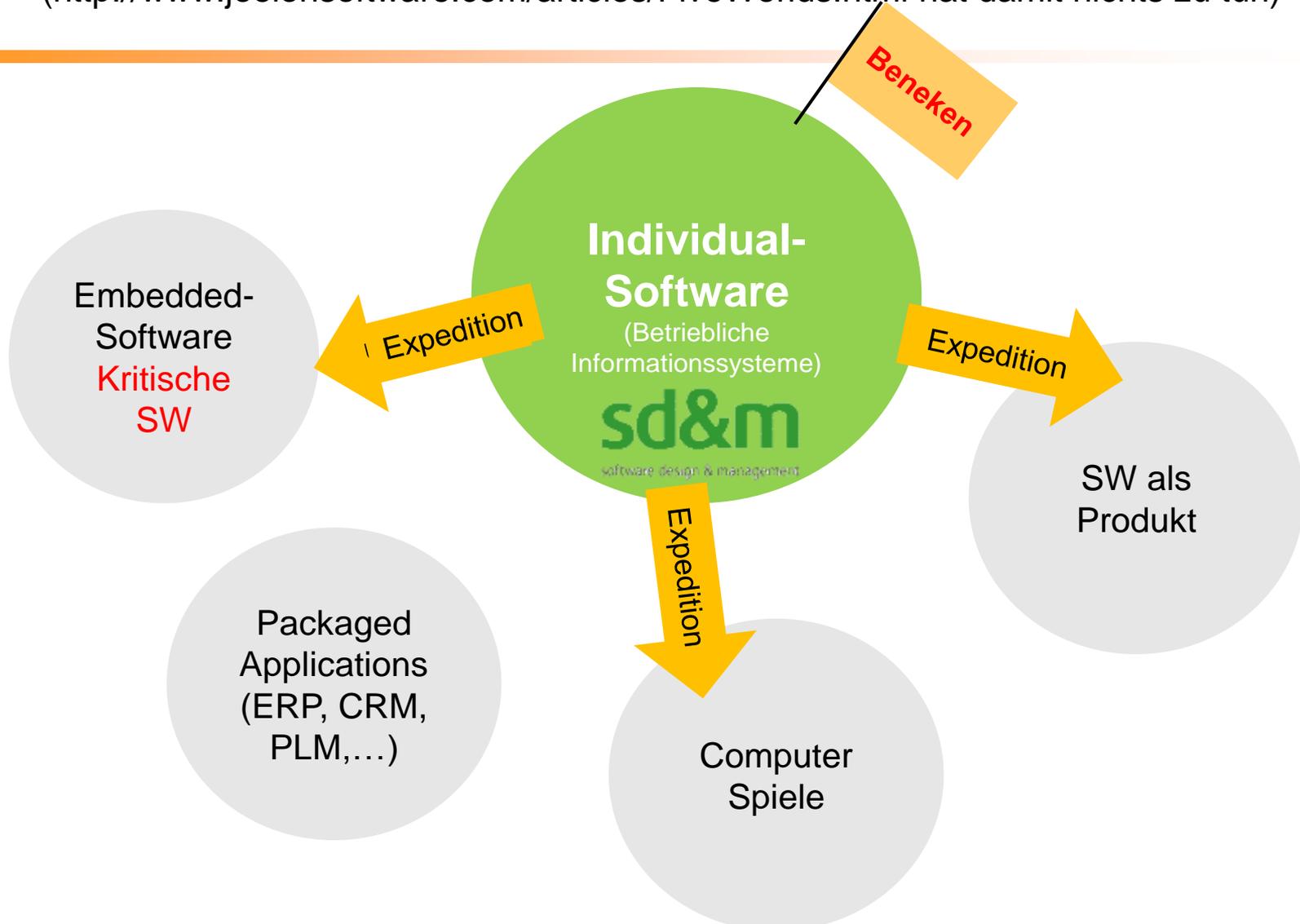
# Heimatplanet: Hochschule Rosenheim

- Hochschule: 5000+ Studierende
  - Bekannt für Holztechnik und Bau
- Studiengänge Inf. (B.Sc./ M.Sc.)+  
Wirtschaftsinf. (B.Sc.):  
500+ Studierende
  - regional
  - ca. 80% FOS / BOS Absolv.
- Rosenheim: Viele IT-Unternehmen  
in Ro. und Umland
  - Organisiert im ROSIK ev.
  - unterstützen FH sehr gut  
(Lehrbeauftragte, Exkursionen,  
Fachvorträge, Projekte, BA/MA ...)
  - Arbeitsmarktlage gerade  
(Herbst 2015) ausgezeichnet



# Heimatplanet: Individual-Software zum Festpreis

(<http://www.joelonsoftware.com/articles/FiveWorlds.html> hat damit nichts zu tun)



# Warum Projekte?

Übung Software-Engineering 1 (ca. 2007)

Einheit über RE und Modellierung



# Projekte im Sommersemester 2015 (Auswahl)

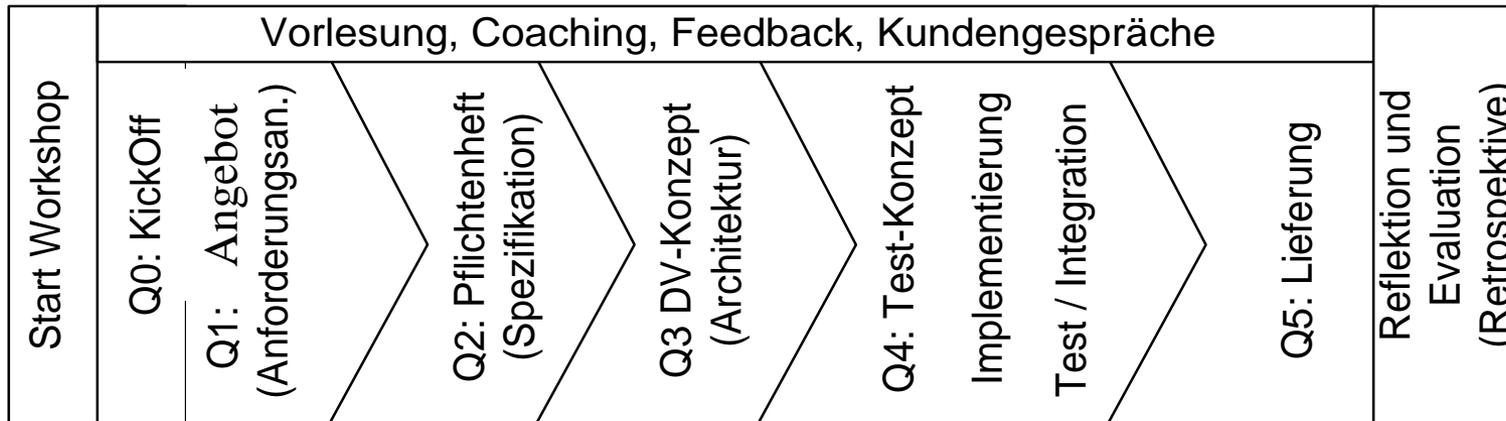


- Kathrein: Prototyp M2M (HW+SW)
- IFT: Akustikmessung im Fensterbau
- Marc O'Polo: Berechtigungssystem für DWH
- Ferchau: Campus App
- IFG: Praktikanten-Datenbank
- jambit: Internes Informationssystem
- TechDivision: Internes System auf Basis Appserver.io
- K&K Prime: Dimmer für LED-Leuchten (SW + HW)
- Diverse F&E Projekte ...

Konzept ist offensichtlich  
erfolgreich

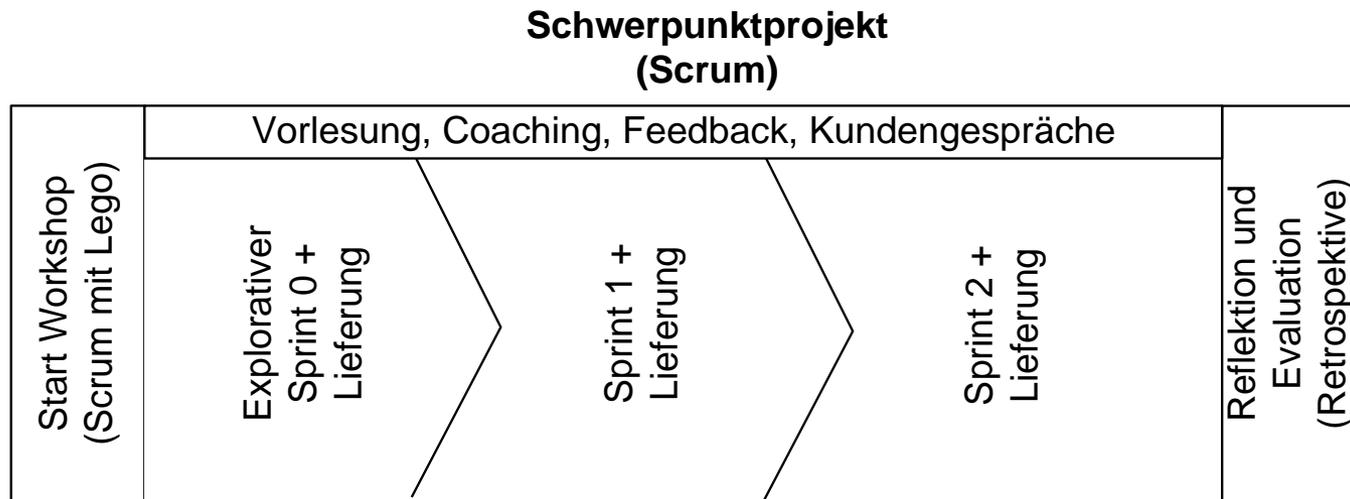
# Software-Engineering 2 Projekte – Konzept

## Software-Engineering 2 (Plan/dokumentgetriebener Wasserfall)



- Projekt wird wie **Festpreis (Werkvertrag)** abgewickelt (aktuell)
- **6 benotete Meilensteine:** Kickoff, Festpreis-Angebot, Pflichtenheft, ...
- Beispiel Angebot (incl. Lastenheftanteile), **Pflichtenheft:** Anwendungsfälle, GUI-Spezifikation, Datenmodell, Qualitätsanforderungen, Randbedingungen
- JourFixe Termin bzw. Vortrag einmal pro Woche, 30-45 Minuten pro Team
- **Wichtig: Viele Vorlagen, Beispiele, enge Begleitung („Kochbuch“)**
- Aufwand: ca. 1-2 Tage pro Woche, Teams mit 3 - 5 Studierenden

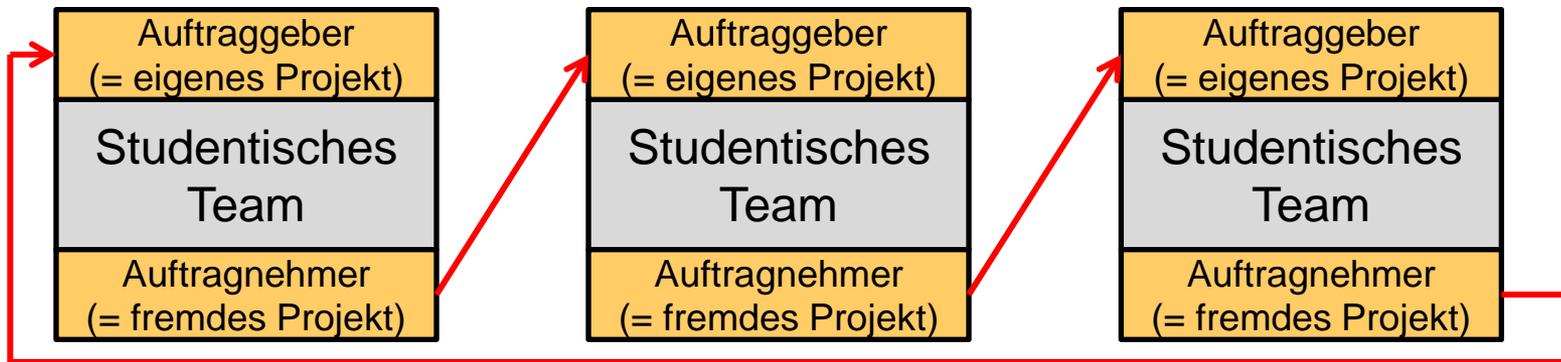
# Schwerpunktprojekte – Konzept



- Projekt wie agile **Produktentwicklung** für Auftraggeber, **Scrum**
- 3 benotete Lieferungen (Sprint Review-Meetings + Retro.)
- RE-Techniken: User Stories, **User Story Maps (!)**, **Wireframes**, ...
- JourFixe Termin bzw. Vortrag einmal pro Woche, 30-45 Minuten
- Aufwand: ca. 1-2 Tage Pro Woche, Teams mit 5 - 7 Studierenden

# Software-Engineering 2, Studenten überlegen *eigene Projekte*

ca. 2003 - 2009



nr	Name	Vorname	Matrikelnummer
1	Schlösser	Hans	123456
2	Mezger	Hugo	123457
3	Müller	Harry	123458

Project details:  
Team-Name (max 20 Zeichen): NukularSoft // se-a-steppenwolf.xls  
E-Mail: stefan-meier@edling.de  
Projektname (max 20 Zeichen): Bibliothek  
Projekt (Kurzbeschreibung max 5 Zeilen): In diesem Projekt geht es um den Verleih und die Rückgabe von Büchern. Eingeschlossen sind Bestandsverwaltung, Kundenverwaltung und Mahnwesen. Abrechnung und Einkauf sind dabei.

- Konzept von Prof. Dr. J. Siedersleben
- Funktioniert sehr gut
- ***Kaum Organisationsaufwand***
- Probleme
  - Requirements-Engineering ***oberflächlich*** und sehr zäh, da häufig ***künstliches Problem*** gelöst wird
  - ***Motivationsprobleme*** bei Spezifikationsphase
  - Software wurde (bei mir) nie ***fertiggestellt***, eher Ruinen, Baustellen und Halbheiten

# Projekte für „echte Kunden“ und was sich dadurch geändert hat ...

- Studierende sind bei „echten“ **wesentlich motivierter** und liefern **wesentlich bessere Ergebnisse**
  - Requirements-Engineering gründlicher (aber nicht gründlich genug)
  - Verständnis von RE-Problemen besser, da Projekt auch gebaut wird (z.B. „Software soll auf jedem Browser laufen ...“)
- **Andere Lehrinhalte wegen (teilweise) gescheiterter Projekte:** weniger Modellieren (keine UML Werkzeuge!), eher praktische Workshop-Durchführung, Umgang mit dem Kunden, Reviews durchführen, mit Issue-Tracker umgehen, SVN/GIT nutzen ...
- Projekte gleichzeitig Mittel um
  - Studierende und Firmen in Kontakt zu bringen
  - Hochschule als Innovationsmotor der Region
  - **Lehre immer wieder mit der Praxis zu konfrontieren**

# Projekte für „echte Kunden“ und was sich dadurch geändert hat ...

- „Echte Projekte“ sind aufwendig für den Dozenten
  - Absprachen mit potentiellen Auftraggebern
  - Kommunikation während des Projektes
  - Abschlusspräsentationen bei erfolgreichen Projekten
  - Letzte Eskalationsebene bei Problemen
- Aufbau von Partnern kostet seine Zeit

**Achtung!**

Dozent(in) ist Coach / Trainer und ist selbst nicht für  
das Projektergebnis verantwortlich

# Projekte der Anfangsphase ca. 2009

Folgerung: Lehrinhalt = **Minimal Marketable Featureset**

## ■ Projektbeispiele

- Software für Messeorganisation für Messe der FH (Sharepoint)
- Hausinterne Suchmaschine (Lucene / Solr)
- Datendarstellung Null-Energie-Stadt
- ...

## ■ **Problem: „Vollständige“ Spezifikation**

- Studenten haben **irgendwelche Anforderungen** aufgenommen, Priorisierung schwierig
- Damit viel zu viele Anwendungsfälle, komplexe GUI und komplexes Datenmodell
- Umgesetzt wurde irgend ein zusammenhangloser Teil
- Software natürlich nicht einsetzbar

## ■ Wichtige Erkenntnis für die nächsten Projekte

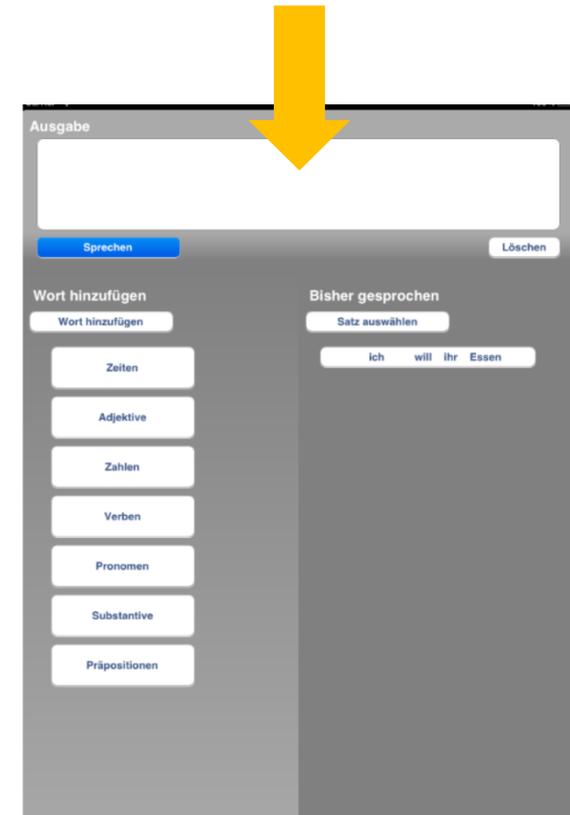
- Klarer Fokus in der Kommunikation: Software **MUSS** später einsetzbar sein, d.h. sie muss etwas Nützliches tun
- Der **vorgegebene Aufwand muss eingehalten werden**
- Zwingend erforderlich: **Sinnvolle Minimierung der Anforderungen**

# Projektbeispiel: TalkerApp

Folgerung: Lehrinhalt = *Usability wesentlich stärker im Fokus (Personas, Usability Test)*



- Ziel: Talker für Kunden mit *Cerebral Parese*
- Durchführung: 4 Informatik-Studenten, ein Jahr
- Ergebnis
  - Erste Prototypen und Projektergebnis nach dem ersten Projekt: Software für Kunden *überhaupt nicht einsetzbar*
  - Großer Erkenntnisgewinn: Annahmen der InformatikerInnen über Kunden häufig falsch
    - Intensive Zusammenarbeit mit Kunden erforderlich, besseres Verständnis wichtig
    - Annahmen und Erkenntnisse über Prototypen ständig überprüfen
- Folgerung: Thema *Gebrauchstauglichkeit* (Usability) ist kaum von RE zu trennen



# Projektbeispiel: FurnitAR

Folgerung: Lehrinhalt = *Minimal Viable Product* (aus Lean Startup)

## ■ Ziele

- Augmented Reality und iPad verbinden: Möbel in eigener Wohnung simulieren
- Kundengruppe: Möbelhersteller

## ■ Durchführung

- 4 Informatik-Studenten
- Projektidee aus Kundenprojekt entstanden

## ■ Ergebnis

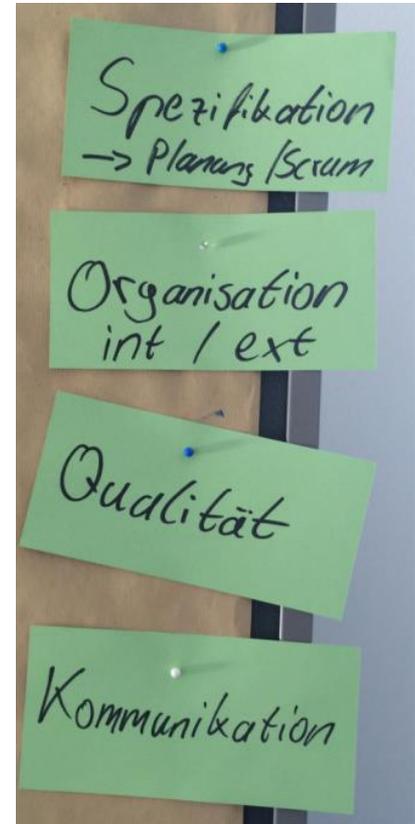
- Technisch sehr cooles, voll funktionsfähiges Produkt
- **Wirtschaftlicher Totalschaden**
- **Grund: Kein Kontakt zu möglichen Kunden, ...**
- Studies und Betreuer haben **angenommen**, dass ein Kunde die Software brauchen könnte, aber es wurde kein „echter“ Kunde befragt, die Annahmen wurden niemals überprüft.



# Projektbeispiel Wissensdatenbank

Folgerung: Lehrinhalt = **Kommunikation mit dem Kunden**,  
**Nichts geht ungeprüft an den Kunden!**

- Ziel
  - Wissensmanagement-System für Beratungshaus in München
- Durchführung
  - 4 Informatik-Studenten, ein Semester
  - **Kommunikationsprobleme** mit dem Kunden (siehe Abbildung aus Touchdown Workshop)
- Ergebnis
  - Team hat Dokumentation und Prototypen nicht rechtzeitig und nicht in ausreichender Qualität geliefert
  - Interne QS hat nicht stattgefunden
  - Software trotzdem produktiv, Softwarequalität war völlig in Ordnung



# Lehrräume sind auch mal Leerräume



- Wichtig: Flexibel konfigurierbarer Lehrraum
- = Alles hat Rollen! Keine fest installierten Rechner
- = Damit: Frontalunterricht, Gruppenarbeit, Open Space, ... möglich
- Materialien: Farbige Post Its, Flip Charts, Whiteboards, Bunte Stifte, ..

# Folgerungen: Fail Better

## ■ Organisation

- RE sehr gut über „echte“ Kundenprojekte vermittelbar (höhere Motivation, viele weitere Ziele damit erreichbar)
- RE Vertiefung dann, wenn die Studierenden konkrete Projekterfahrung haben
- Bewährt: Festpreis und dokumentgetriebener Wasserfall vs. Produktentwicklung mit Scrum
- Richtiger Lehrraum wichtig für Projekterfolg und flexible Lehre

## ■ Inhalte

- RE das wichtigste Thema im Software-Engineering, da es den Projektinhalt definiert
- Scoping: Minimalität als zentrales Konzept im RE. Aktuell in Rosenheim „Minimal Viable Product“
- Usability: untrennbar mit RE verbunden
- Praktisches RE wichtiger als Theorie
  - Inhalte eher Workshops durchführen, Kundenkomm., Reviews, ...
  - Als UML / Modellierung, Vorgehensmodelle, RE-Frameworks, ...

# Studienziel im Bachelor-Informatik

## Software – Ingenieur(in)

*Ab dem ersten Tag produktiv einsetzbar*

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

# Kontaktdaten

**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](mailto:gerd.beneken@fh-rosenheim.de)

Hochschule **Rosenheim**  
University of Applied Sciences

