

**5CS-SEPM-40**

# **Projektmanagement**

## **Teil 1 - Einführung**

**Dr. Jörg Härtwig / Dr. Kyrill Meyer**

- Wer sind die Dozenten?
- Worum geht es in diesem Teil des Moduls?
- Wie wird gearbeitet und was kommt auf mich zu?



- Kennenlernen
- Eine grobe Vorstellung über die Inhalte des Kurses entwickeln
- Eine Arbeitsvereinbarung treffen

## Dr. Kyrill Meyer

- Studium der Informatik
- Promotion 2009 zum DR. RERUM NATURALIUM  
Forscher und Projektmanager an der  
Universität Leipzig bis 2015
- Geschäftsführer und Abteilungsdirektor des  
Instituts für Angewandte Informatik bis 2018
- Gründer IFDT / GFDT  
Institut für Digitale Technologien gGmbH  
2017 – heute

## Dr. Jörg Härtwig

- Studium der Informatik
- Promotion 2008 zum DR. RERUM NATURALIUM
- Leiter der Softwareentwicklung
  - von 2008 – 2010 IF Software Services
- Geschäftsführer IF Software Services
  - von 2010 – 2011
- Vorstand (CTO) INTER-FORUM AG
  - von 2011 – 2015
- Geschäftsführer (CTO) INTER-FORUM GmbH
  - von 2015 – 2017
- Gründer IFDT  
Institut für Digitale Technologien gGmbH  
2017 – heute



**IFDT** Institut für  
Digitale Technologien



**GFDT** Gesellschaft  
für Digitale Technologien

## Agenda

# PROJEKTMANAGEMENT IM MODUL SOFTWARETECHNIK UND PROJEKTMANAGEMENT

Werden im Laufe der Vorlesung noch konkretisiert

- 180 Minuten Prüfung am Ende des Semesters
  - 120 min SE am Computer
  - 60 min PM schriftlich
- Fünf bis zehn Fragen zu Theorie und Praxis

Die Studierenden lernen die UML-Notation zur Modellierung von Software-Architekturen und Anwendungsdomänen kennen. Meta-Modelle sind ebenso Bestandteil des Moduls wie CASEWerkzeuge und Software-Muster als Design-Technik. **Die Studierenden lernen, Projekte selbstständig zu bearbeiten.** Sie werden befähigt, während der Entwicklung den kompletten Software-Lebenszyklus unter Anwendung eines passenden Vorgehensmodells zu durchlaufen.

Können
Instrumentale Kompetenz
Durch ihr Wissen sind die Studierenden in der Lage, wohl durchdachte und klar strukturierte Anwendungssysteme zu entwerfen und als Quelltext unter Verwendung entsprechender CASE-Werkzeuge generieren zu lassen. Sie können Software-Projekte über alle Phasen des Entwicklungszyklus' hinweg bearbeiten und steuern.
Systemische Kompetenz
Die Studierenden besitzen die notwendige Kompetenz, um einen Software-Entwicklungsprozess mit modernen CASE-Werkzeugen zu begleiten und zu gestalten. Sie erfassen die Bedeutung von Projekt- und Qualitätsmanagement für ein Unternehmen, sowohl im täglichen Geschäft als auch im Marktumfeld des Unternehmens.
Kommunikative Kompetenz
Nicht zuletzt durch die erlernte Notation der UML-Diagramme ist eine deutlich verbesserte Kommunikationsfähigkeit der Studierenden mit allen Projektbeteiligten (Anwendern, Fachexperten, Software-Architekten) gegeben. Die Studierenden sind in der Lage, sich gegenüber Fachvertretern und Laien sowohl mündlich als auch schriftlich auf fachlich korrekte Art und Weise zu Themen des Projektmanagements zu äußern.

# Aufgabe 1 – Was Sie mitbringen

Schreiben Sie auf einen Zettel

- Was bedeutet Projektmanagement für Sie?
- Welche drei Begriffe verbinden Sie mit dem Thema Projektmanagement?
- Welche neuen Erkenntnisse wollen Sie aus der Vorlesung mitnehmen?

Die Fragen werden anschließend gemeinsam ausgewertet.



- Theoretisch die wichtigsten Inhalte klären
  - Begriffe und Konzepte
  - Lebenszyklus, Aufgaben und Rollen im Projektmanagement
- Praktisch in die Anwendung gehen
  - Projektpläne erstellen
  - Ressourcen zuweisen
  - Dokumentieren
- Querverbindungen zu
  - Softwarequalität und Software Engineering Prozessen
  - Rechtliche, Marketing und Finanzierungsaspekte

## frontal klassisch

- Wissensvermittlung linear
- Studierende sind nicht involviert
- im Regelfall keine Fragen stellen
- im Regelfall keine Übungen
- Übungen finden in Seminaren statt

## interaktiv agil

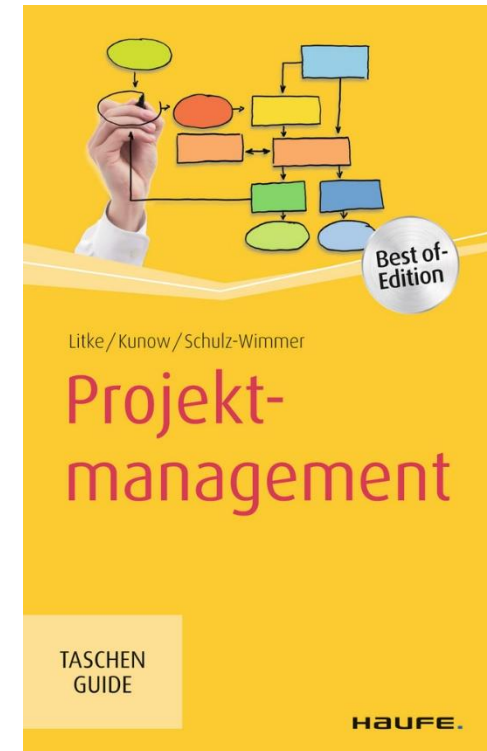
- Wissensvermittlung Querschnitt
- Studierende sind involviert
- im Regelfall Fragen stellen
- Übungen sind Teil der Veranstaltung

Projektmanagement bietet sich für den interaktiven und agilen Stil an.

- Vorlesung mit praktischen Anteilen
- Mitwirkung des Einzelnen
- Fragen erlaubt und erwünscht
- Offenheit für verschiedene Meinungen



Axel Buhl. Grundkurs Software-Projektmanagement. München: Hanser, 2004



Hans-D. Litke, Ilonka Kunow, Heinz Schulz-Wimmer  
Projektmanagement, Haufe, 2018

## Projektmanagement:

- Matthias Geirhos: IT-Projektmanagement: Was wirklich funktioniert – und was nicht. Ratgeber IT-Projektleiter. Bonn: Rheinwerk Verlag, 2017
- Pascal Mangold: IT-Projektmanagement kompakt. Heidelberg: Spektrum Akademischer Verlag, 2009
- Harold Kerzner. Projektmanagement. mitp, 2008
- Roman Heimbold. Endlich im grünen Bereich! Projektmanagement für Jedermann. mitp, 2005
- Uwe Braehmer. Projektmanagement für kleine und mittlere Unternehmen. Schnelle Resultate mit knappen Ressourcen. Hanser Wirtschaft, 2005
- Axel Buhl. Grundkurs Software-Projektmanagement. München: Hanser, 2004
- Zukunft im Projektmanagement. Editors: Gerrit Kerber et al. Glashütten: dpunkt, 2003
- Harry M. Sneed. Software-Produktmanagement. Heidelberg: dpunkt, 2005
- Carl August Zehnder. Informatik-Projektentwicklung. Zürich: vdf, 2003

## Soft Skills:

- Tomas Bohinc. Projektmanagement. Soft Skills für Projektleiter. Gabal, 2006
- Uwe Vigenschow. Soft Skills für Softwareentwickler. Heidelberg: dpunkt, 2007
- Michael Prokop. Open Source Projektmanagement. München: Open Source Press, 2010

## Software Engineering Process (SEP):

- Chris Rupp. Requirements-Engineering und -Management. Professionelle, iterative Anforderungsanalyse für die Praxis. Hanser Fachbuchverlag, 2006
- Andreas Essigkrug. Rational Unified Process kompakt. Spektrum Akademischer Verlag, 2007
- Gerhard Versteegen. Projektmanagement mit dem Rational Unified Process. Berlin: Springer, 2000
- Kent Beck. Extreme Programming Explained: Embrace Change. Addison-Wesley Longman, 2005
- Henning Wolf. eXtreme Programming. dpunkt, 2005
- Richard Hightower. Professional Java Tools for XP. Ant, XDoclet, JUnit, Cactus, and Maven. Wiley Publishing, 2004
- Frank Westphal. Testgetriebene Entwicklung mit JUnit & FIT. Wie Software änderbar bleibt. dpunkt, 2005
- Alistair Cockburn. Agile Software Development. The Cooperative Game. Addison-Wesley Longman, 2006
- Wolfgang Cronenbroeck. Standards und Vorgehensmodelle in internationalen Projekten. GULP Knowledge Base, Juli 2008. <http://www.gulp.de>

## Quality Assurance:

- Georg Erwin Thaller. ISO 9001:2000. Software-Entwicklung in der Praxis. Hannover: Heise, 2001
- Heinrich Hübscher. IT-Handbuch. Braunschweig: Westermann, 2001
- Marc C. Paulk. The Capability Maturity Model for Software. Guidelines for Improving the Software Process. (SEI Series in Software Engineering.) Addison Wesley, 1995
- Peter Liggesmeyer. Software-Qualität. Testen, Analysieren und Verifizieren. Heidelberg: Spektrum, 2002
- Andreas Spillner. Basiswissen Softwaretest. dpunkt, 2005
- Harry M. Sneed. Der Systemtest. Anforderungsbasiertes Testen von Software-Systemen. Hanser, 2006
- Johannes Link. Softwaretests mit JUnit. dpunkt, 2005



- Wir danken Herrn Prof. Dr.-Ing. Christian Heller für die Bereitstellung von Material, welches in die Ausarbeitung der Vorlesung eingeflossen ist.