

Software-Engineering Wintersemester 2000/2001

Lehrstuhl für Softwaretechnologie, Prof. Dr. Willem-Paul de Roever

Zeit: Mi. 8 ct (3 stündig)

Ort: n. V.

Fachgebiet: praktische Informatik

Zielgruppe: Studierende im Hauptstudium

Die Vorlesungs- und Übungsteile dieser Vorlesung formen ein integriertes Angebot. Software-Engineering erlernt man (1) durch das Ausführen eines Software-Projektes gem. den Phasen des Software-Life-Cycles und (2) während des Projektes durch die Definition und Ausarbeitung der Schnittstellen zur Arbeit anderer Teams. Daher wird die Theorie des Software-Engineerings im Vorlesungsteil behandelt und die Praxis in den Übungen. Als Ergänzung wird empfohlen, die Vorlesung *Objektorientierte Programmierung* von Herrn Simon zzu hören (die Inhalte sind aufeinander abgestimmt).

Im Vorlesungsteil werden die folgenden Themen behandelt:

Teil 1 An Hand moderner Konzepte der objektorientierten Software-Entwicklung wird in Teil 1 eine Einführung in den Software-Life-Cycle geboten, und zwar:

- der Software-Prozess
- Software-Life-Cycle Modelle
- Testing
- Einführung in das objektorientierte Design
- UML: use cases, class diagrams, interaction diagrams ...
- Planung, Teams, usw.

Teil 2 behandelt die unterschiedlichen Phasen des Software-Life-Cycles:

- Anforderungsphase
- Spezifikationsphase
- objektorientierte Analysephase
- Designphase
- Implementierungs- und Integrationsphase
- Software-Wartungsphase

Teil 3 Software Standards: ISO 9000, Capability Maturity Modell (CMM).

Teil 4: Software-Engineering und formale Methoden Besteht aus einer Einführung in das graphische Software-Modellierungssystem StateMate mit entsprechender Praktikumsbegleitung und Projektbeteiligung.

Die Eigenschaften des StateMate Systems (Simulation, Verifikation) ermöglichen es, den Software-Life-Cycle bei der Anwendung von StateMate zu verkürzen.

Während der Vorlesung wird das u.a. Buch von Stephen Schach [9] als Leitfaden benutzt.

Weitere Informationen zur Vorlesung unter:

<http://www.informatik.uni-kiel.de/inf/deRoeever/WS0001/SoftwareEngineering>

Literatur

- [1] Timothy Budd. *An Introduction to Object-Oriented Programming*. Addison-Wesley, 2 edition, 1997.
- [2] D. Mandrioli C. Ghezzi, M. Jazayeri. *Fundamentals of Software Engineering*. Prentice Hall, 1991.
- [3] Martin Fowler and Kendall Scott. *UML Distilled: Applying the Standard Object Modeling Language*. Addison-Wesley, 1997.
- [4] H. Gomaa. *Software Design Methods for Concurrent and Real-Time Systems*. Addison-Wesley, 1993.
- [5] David Harel and Michal Politi. *Modeling Reactive Systems with Statecharts; the STATEMATE Approach*. McGraw-Hill, 1998.
- [6] Philippe Kruchten. *The Rational Unified Process*. Addison-Wesley, 1998.
- [7] Bertrand Meyer. *Objektorientierte Softwareentwicklung*. Hauser Verlag, 1990.
- [8] J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, and W. Lorenzen. *Objektorientiertes Modellieren und Entwerfen*. Hauser Verlag, 1993.
- [9] S. R. Schach. *Classical and Object-Oriented Software Engineering*. Irwin, 3 edition, 1996.
- [10] Stephen. R. Schach. *Classical and Object-Oriented Software Engineering with UML and Java*. McGraw-Hill, 4 edition, 1999.
- [11] Ian Sommerville. *Software Engineering*. Addison-Wesley, 5th edition, 1998.
- [12] Hans van Vliet. *Software Engineering: Principles & Practice*. Wiley, 2nd edition, 1993.