

Kommentiertes Literaturverzeichnis

Balzert, H. (1996). *Lehrbuch der Software-Technik: Software-Entwicklung*. Heidelberg: Spektrum Akademischer Verlag. 1009 p. + CD-ROM.

Balzert H. (1998). *Lehrbuch der Software-Technik: Software-Management, Software-Qualitätssicherung, Unternehmensmodellierung*. 769 p. + CD-ROM.

Enzyklopädische Fülle von Material. Präsentation überladen mit Details. Konzepte, Zusammenhänge und die Bedeutung der einzelnen Themen fehlen oder gehen unter. Behandelt auch viele nicht zum Kern des Software Engineerings gehörende Themen. Als Lehrbuch weniger geeignet.

Boehm, B. (1981). *Software Engineering Economics*. Englewood Cliffs, N.J.: Prentice Hall. 767 p.

Ein Standardwerk. Fundgrube für empirische und quantitative Aussagen. Als Lehrbuch weniger geeignet.

Boehm, B. (1983). Seven Basic Principles of Software Engineering. *Journal of Systems and Software* **3**, 3-24.

Ein klassischer Zeitschriften-Artikel, welcher v.a. den Management-Aspekt des Software Engineerings in sieben Prinzipien zusammengefasst darstellt.

Brössler, P., J. Siedersleben (Hrsg) (2000). *Softwaretechnik: Praxiswissen für Software-Ingenieure*. München: Hanser. 323 p.

Sammlung von Aufsätzen über die systematische Entwicklung von Software-Systemen in der Praxis eines großen deutschen Softwarehauses.

Brooks, F.P. (1995). *The Mythical Man Month. Essays on Software Engineering*. Anniversary Edition Reading, Mass., etc.: Addison-Wesley.

Unterhaltsame Beschreibung von Fehlern im Software Engineering und ihren Ursachen. Neuausgabe des Originals von 1975 mit vier neuen Kapiteln. Eine Pflichtlektüre für alle, die sich für Software Engineering interessieren.

Brooks, F.P. (1987). No Silver Bullet. Essence and Accidents of Software Engineering. *IEEE Computer* **20**, 4. 10-19.

Klassischer Zeitschriftenartikel über Einfaches und Schwieriges (und die Nichtexistenz von Patentlösungen) im Software Engineering. Auch zu finden in Brooks (1995).

Denert, E. (1991). *Software Engineering*. Berlin, etc.:Springer. 452 p.

Kein Buch über Software Engineering allgemein, sondern ein Lehrbuch über Entwicklung von Informationssystemen nach einer vom Verfasser geschaffenen Methode. Für dieses Teilgebiet ein sehr gutes, praxisnahes Buch; als allgemeine Einführung in Software Engineering nicht geeignet.

Dumke, R. (1993). *Modernes Software Engineering: Eine Einführung*. Braunschweig/ Wiesbaden: Vieweg. 305 p.

Sammelsurium verschiedener Themen aus dem Software Engineering. Nicht empfehlenswert.

Endres, A., D. Rombach (2003). *A Handbook of Software and Systems Engineering*. Harlow, England, etc.: Pearson Addison-Wesley. 327 p.

Trägt das empirisch erhärtete Wissen über Software Engineering systematisch zusammen. Als ergänzende Lektüre hervorragend, als eigenständiges Lehrbuch weniger geeignet.

Fairley, R. (1985). *Software Engineering Concepts*. New York: McGrawHill. 364 p.

Kompakte, sehr gute Einführung in die Thematik. Leider seit Erscheinen nicht mehr revidiert, daher nicht mehr auf dem neuesten Stand, z.B. keine Aussagen über Objektorientierung. Trotzdem immer noch lesenswert.

Frühauf, K., Ludewig, J., Sandmayr, H. (1999). *Software-Projektmanagement und -Qualitätssicherung*. Dritte, überarbeitete Auflage. Zürich: vdf Hochschulverlag. 170 p.

Einführung in zwei zentrale Teilgebiete des Software Engineerings in der Art einer Fibel. Empfehlenswert.

Ghezzi C., M. Jazayeri, D. Mandrioli (2002). *Fundamentals of Software Engineering*. Second Edition. Englewood Cliffs, N.J.: Prentice Hall. 624 p.

Ansprechende Darstellung. Gewicht mehr auf technischer Seite des SE, Beschreibung nicht nur von Ausprägungen, sondern auch der zugrundeliegenden Konzepte. Empfehlenswert.

Gilb, T. (1988). *Principles of Software Engineering Management*. Reading, Mass.: Addison Wesley. 442 p.

Baut auf drei Ideen auf: Messbare Anforderungen, evolutionäre Entwicklung, Prüfung mit Inspektionen. Keine vollständige Einführung ins Thema Software Engineering. Populistischer Stil in der Darstellung.

Jalote, P. (1997). *An Integrated Approach to Software Engineering*. Second edition. Berlin, etc.: Springer. 497 p.

Konventionelle Einführung anhand eines Durchgangs durch den Software Lebenslauf. In der zweiten Auflage sind zu den strukturierten Methoden die objektorientierten hinzugekommen. Durchgehende Fallstudie.

Johnson, J.R. (1991). *The Software Factory: Managing Software Development and Maintenance*. 2nd edition. Wellesley, Mass.: QED Information Sciences. 277 p.

Etwas oberflächlich/populärwissenschaftlich; Themenauswahl stark auf Entwicklung betrieblicher Informationssysteme bezogen. Stärke: Enthält in den Anhängen Zahlen zu Messungen im Software-Entwicklungsprozess.

Jones, G.W. (1990). *Software Engineering*. New York: John Wiley & Sons. 480 p.

Vom Inhalt her an sich kein schlechtes Buch. Insgesamt aber ist das Material zu breit und zu oberflächlich dargestellt. Schlechtes, das Lesen behinderndes Layout.

Kahlbrandt, B. (1998). *Software-Engineering*. Springer, Berlin, etc.

Etwas unsystematische Mischung aus Lehrbuch über Software Engineering und Einführung in die UML.

Kimm, R. et al. (1979). *Einführung in Software Engineering*. Berlin: Walter de Gruyter. 306 p.

Zum Zeitpunkt seines Erscheinens eines der besten Lehrbücher zum Thema. Aus heutiger Sicht teilweise hoffnungslos veraltet, teilweise mit erheblichen Lücken. Heute als Lehrbuch leider nicht mehr verwendbar.

Lamb, D.A. (1988). *Software Engineering: Planning for Change*. Englewood Cliffs, N.J.: Prentice Hall. 298 p.

Knappe, teilweise oberflächliche Behandlung einer Sammlung von SE-Themen. Schwerpunkt auf Spezifikation und Entwurf. Grundprobleme und Zusammenhänge werden nicht verdeutlicht.

Ludewig, J. (Hrsg.) (1991). *Software- und Automatisierungsprojekte - Beispiele aus der Praxis*. Stuttgart: Teubner. 227 p.

Kein Lehrbuch, sondern eine Zusammenstellung von Erfahrungsberichten über abgewickelte Software-Projekte aus der Praxis. Besonders für Studierende ein guter Einblick in Anwendung bzw. Nicht-Anwendung von Software Engineering-Prinzipien in der Praxis. Nicht mehr im Buchhandel erhältlich.

Marco, A. (1990). *Software Engineering*. Englewood Cliffs, N.J.: Prentice Hall. 544 p.

Darstellung etwas breit, Themenauswahl im Detail z.T. etwas eigenwillig, keine klare und systematische Darstellung der Grundprinzipien.

Mayrhauser, A. von (1990). *Software Engineering - Methods and Management*. San Diego: Academic Press. 864 p.

In Inhalt und Aufbau von mäßiger Qualität, teilweise veraltet, viel zu umfangreich: nicht empfehlenswert.

McDermid, J. (1991). *Software Engineer's Reference Book*. Oxford: Butterworth-Heinemann. ca. 1200 p.

Behandelt alle Gebiete der Informatik mit Bezug zum Software Engineering einschließlich der Grundlagen- und Randgebiete in knappen Übersichtsartikeln in der Art einer Enzyklopädie, aber nach Sachgebieten, nicht nach Stichworten geordnet. Nachschlagewerk, kein Lehrbuch.

Nagl, M. (1990). *Softwaretechnik: Methodisches Programmieren im Großen*. Berlin, etc.: Springer. 387 p.

Beschreibt im Wesentlichen nur das Problem des Software-Entwurfs; dieses aber sorgfältig und gründlich. Keine Einführung in Software Engineering.

Pagel, B.U., H.W. Six (1994). *Software Engineering - Band 1: Die Phasen der Software-Entwicklung*. Bonn, etc.: Addison-Wesley. 893 p.

Beschreibt in diesem ersten Band ausschließlich die Entwicklungstätigkeiten. Diese sind recht breit dargestellt.

Pfleeger, S.L. (1991). *Software Engineering: The Production of Quality Software*. 2nd edition. New York: Macmillan Publishing. 517 p.

Leitfadenartig aufgebaut, Auswahl und Darstellung vor allem im Detail oberflächlich, unsystematisch und teilweise veraltet.

Pomberger, G., G. Blaschek (1993). *Grundlagen des Software Engineering: Prototyping und objektorientierte Software-Entwicklung*. München: Hanser. 337 p.

Beschreibt die Arbeit in den einzelnen Entwicklungsphasen, ca. ein Drittel des Buchs behandelt Objektorientierung, wenig über den Management-Aspekt des Software Engineerings.

Pressman, R.S. (1992). *Software Engineering - A Practitioner's Approach*. Third Edition. New York, etc.: McGrawHill. 793 p.

Ansprechende und praxisorientierte, aber sehr umfangreiche Darstellung des Themas. Vor allem die technischen Aspekte sind sehr breit dargestellt.

Schach, S.R. (1996). *Classical and Object-Oriented Software Engineering*. Third edition. Chicago, etc: IRWIN. 603 p.

Gute Darstellung der Entwicklungsphasen. Objektorientierung wurde in der dritten Auflage hinzugefügt (die zweite Auflage hieß noch schlicht "Software Engineering"), was der Darstellung teilweise noch anzumerken ist. Management-Aspekte des Software Engineerings werden eher oberflächlich behandelt. Sehr gute Verweise auf weiterführende Literatur.

Schach, S.R. (1999). *Classical and Object-Oriented Software Engineering with UML and Java*. Fourth edition. Boston: WCB/McGraw-Hill. 616 p.

Teilweise reorganisierte, leicht erweiterte und aktualisierte Fassung von Schach (1996). Beispiele neu mit UML und Java. *Keine* Einführung in UML bzw. Java.

Sommerville, I. (2004). *Software Engineering*, Seventh Edition. Pearson Education. 963 p. (deutsche Übersetzung der sechsten Auflage: *Software Engineering*, München: Pearson Studium (Addison Wesley), 2001).

Umfassende Darstellung des Themas. In der Fülle ist jedoch das Wesentliche nicht mehr überall gut zu erkennen. Darstellung trotz ihres Umfangs zum Teil oberflächlich. In vielen Bibliotheken sind auch ältere Auflagen noch greifbar.

Suhr, R., R. Suhr (1993). *Software Engineering: Technik und Methodik*. München, Wien: Oldenbourg. 415 p.

Darstellung verschiedener, schwergewichtig technischer Aspekte des Software Engineerings. Ansprechend im Aufbau, leider in der Ausführung vieler Themen unbefriedigend.

van Vliet, H. (1993). *Software Engineering: Principles and Practice*. Chichester: John Wiley & Sons. 558 p.

Übersicht über alle Themengebiete des Software Engineerings, vielfach jedoch sehr oberflächlich. Die dargestellten Techniken sind teilweise veraltet, neuere fehlen.

Zehnder, C.A. (2001). *Informatik-Projektentwicklung*. 3. Auflage. Zürich: vdf Hochschulverlag. 327 p.

Konzentriert sich auf den Aspekt der Software-Entwicklung nach einem bestimmten Entwicklungsmodell. Beschreibt vor allem, was zu tun ist; wenig Angaben zur Methodik.

Zuser, W., S. Biffel, T. Grechenig, M. Köhle (2001). *Software Engineering mit der UML und dem Unified Process*. München: Pearson Studium. 377 p.

Kein Lehrbuch, sondern ein Leitfaden für Praktiker, der auf Software-Projektentwicklung und die Verwendung von UML sowie des Unified Process der Firma Rational fokussiert ist. Da die konzeptionelle Seite dabei zu kurz kommt, ist das Buch als Einstiegslektüre für Studierende weniger geeignet.