
Inhalt und Aufbau, Zielgruppen

Inhalt

Vier Kriterien entscheiden darüber, welche Themen in einem Lehrbuch behandelt werden:

- Welches Wissen bringen die Autoren mit?
- Welche Themen sind für die Leser wichtig und attraktiv?
- Welche Aussagen und Erkenntnisse sind mutmaßlich für einige Jahre stabil, nicht den kurzfristigen Moden unterworfen?
- Was kann und sollte in einer Vorlesung über Software Engineering behandelt werden?

Jedes der Kriterien definiert, mehr oder minder scharf, eine Menge; die Schnittmenge liefert den Themenkatalog, der dem Buch zu Grunde liegt.

Dabei müssen auch – wie immer im Software Engineering – Kompromisse gefunden, also Einschränkungen beim einen oder anderen Kriterium in Kauf genommen werden. Für die Leser sollte aber in allen Teilen deutlich werden, welches Verständnis wir von unserem Fach haben: Software Engineering ist eine auf der Informatik beruhende Ingenieurdisziplin, die wie alle Ingenieurfächer darauf abzielt, die Praxis zu verbessern.

Software Engineering stellt sich für uns wie ein weitgehend unerforschter Kontinent dar. Wir sind weit davon entfernt, eine vollständige und präzise Beschreibung anbieten zu können. Wenigstens aber die Konturen, die bisherige Entwicklung (in groben Zügen) und die als gesichert geltenden Einsichten soll dieses Buch abdecken. Wer es in der Lehre einsetzt, wird eigene Erfahrungen und spezielle Literatur hinzufügen.

Aufbau

Für den Aufbau eines Lehrbuches gibt es einige sinnvolle Prinzipien:

- vom Allgemeinen zum Speziellen
- vom Leichten zum Schwierigen
- von den Grundlagen zu den Anwendungen

Im Software Engineering kommt noch hinzu:

- dem Gang des Software-Projekts folgend

Leider lässt sich daraus keine konkrete Reihenfolge ableiten, denn diese Prinzipien haben unterschiedliche Konsequenzen. Zudem sind viele Themen zyklisch verbunden. Ein typisches Beispiel ist die Software-Wartung: Im Projektablauf steht sie ganz am Ende. Da aber die Schwierigkeiten der Wartung sehr stark von den Entscheidungen bei der Konstruktion der Software beeinflusst sind, sollten Überlegungen zur Wartung nicht erst angestellt werden, wenn die Entwicklung abgeschlossen ist, sondern bereits in der Projektplanung. Die Wartung wirkt sich also auf den Entwurf aus, weil sich der Entwurf auf die Wartung auswirkt.

Wir können dieses Dilemma nicht auflösen. Wir haben darum einen anderen Aufbau gewählt: In sechs Teilen wird das Thema aus verschiedenen Perspektiven betrachtet.

- Teil I: *Grundlagen*

Die Inhalte dieses Teils sind fundamental und damit nicht von einer speziellen Technologie geprägt. Hier geht es um das Basiswissen des Software-Ingenieurs. Am Anfang steht ein Kapitel über Modelle, weil dieses Thema für das Software Engineering wirklich grundlegend ist und damit das Fundament aller weiteren Kapitel darstellt.

- Teil II: *Menschen und Prozesse*

Software ist enger als andere technische Artefakte mit dem Denken der Menschen verknüpft, die die Software herstellen oder benutzen. Die organisatorischen Rahmenbedingungen haben großen Einfluss auf den Erfolg der Projekte. Diese Punkte werden im Teil II behandelt.

- Teil III: *Daueraufgaben im Software-Projekt*

Viele Tätigkeiten wie Dokumentation oder Prüfung können nicht einzelnen Schritten der Entwicklung oder speziellen Dokumenten zugeordnet werden, sie finden laufend statt. Teil III behandelt diese Daueraufgaben.

- Teil IV: *Techniken der Software-Bearbeitung*

Die einzelnen Schritte der Entwicklung, von der Analyse bis zur Integration, werden im Teil IV erläutert.

■ Teil V: *Verwaltung und Erhaltung der Software*

Die Wartung ist eng verknüpft mit der Konfigurationsverwaltung, dem Reengineering und der Wiederverwendung. Darum werden alle diese Themen im Teil V des Buches behandelt. Untereinander ist die Abgrenzung unklar; beispielsweise wird die Änderungsverwaltung im Kapitel über die Wartung besprochen, sie könnte ebenso gut im Kontext der Konfigurationsverwaltung behandelt werden.

Die Themen dieses Teils könnten auch dem Teil III, den Daueraufgaben, zugeordnet werden. Hier handelt es sich aber um Aufgaben und Arbeiten, die ganz oder vorwiegend anfallen, nachdem die Software an den Kunden ausgeliefert ist.

■ Teil VI: *Nachwort, Literatur und Index*

Bis zum Teil V geht es um die Frage, wie Software in der Praxis bearbeitet werden sollte. Der Teil VI enthält eine Reflektion darüber, wie die Lage im Lehr- und Forschungsgebiet Software Engineering einzuschätzen ist und wie sie sich nach Meinung der Autoren verändern wird.

Die zitierte Literatur haben wir nach Verfassern, die zitierten Normen nach Normenreihen geordnet; wir haben uns sehr nachdrücklich bemüht, alle Quellen richtig und vollständig anzugeben. Das Stichwortverzeichnis steht ganz am Schluss des Buches.

Wer dieses Buch im Unterricht einsetzt, sollte das Inhaltsverzeichnis nicht als Vorgabe für die Gliederung seiner Lehrveranstaltung betrachten. Wir hoffen, unsere Kolleginnen und Kollegen durch eine klare und nachvollziehbare Struktur bei der individuellen Auswahl der Themen und bei der Gestaltung ihrer Lehre zu unterstützen. Und natürlich gibt es Bedarf und Raum für Ergänzungen durch andere Themen, die in diesem Buch fehlen oder nur kurz besprochen werden. Die formale Spezifikation und eine ganze Reihe moderner Entwicklungstechnologien sind naheliegende Beispiele solcher Gebiete.

Zielgruppen

Da wir regelmäßig Lehrveranstaltungen über Software Engineering an zwei deutschen Universitäten durchführen, sind unsere Studenten die Adressaten dieses Buches. Unsere Erfahrungen in anderen Ländern und in anderen Lehranstalten legen die Vermutung nahe, dass das Buch für alle deutschsprachigen Länder und auch für andere Hochschulen geeignet ist. Dazu zählen wir auch Schulungseinrichtungen in der Industrie, wo wir selbst etwa zwei Jahrzehnte lang immer wieder gern unterrichtet haben.

Aus dieser Erfahrung wissen wir, dass es in der Industrie sehr viele Menschen gibt, die an Software arbeiten, ohne eine einschlägige Ausbildung zu haben. Die meisten von ihnen sind gelernte Ingenieure, Naturwissenschaftler, Mathematiker oder Kaufleute; sie bringen durch ihren ursprünglichen Beruf wichtige Vorausset-

zungen für das Software Engineering mit und haben Studenten sehr viel praktische Erfahrung voraus. Dieses Buch gibt ihnen die Möglichkeit, einige Grundlagen und spezielle Themen im Selbststudium zu erarbeiten. Wir haben uns besonders für diese Gruppe darum bemüht, die Ideen und Gedanken nicht nur aufzulisten, sondern durch einen hoffentlich gut strukturierten und formulierten Text auch verständlich und angenehm lesbar zu machen, also quasi eine Vorlesung in Buchform anzubieten.