



Software Engineering

10., aktualisierte Auflage

Ian Sommerville

 Pearson

 EXTRAS
ONLINE

Software Engineering

10., aktualisierte Auflage

Ian Sommerville

Software Engineering - PDF

Inhaltsverzeichnis

Software Engineering

Impressum

Inhaltsübersicht

Inhaltsverzeichnis

Vorwort

Teil I - Einführung in Software-Engineering

Kapitel 1 - Einführung

1.1 Professionelle Softwareentwicklung

1.1.1 Software-Engineering

1.1.2 Vielfalt des Software-Engineerings

1.1.3 Software-Engineering und das Internet

1.2 Ethik des Software-Engineerings

1.3 Fallstudien

1.3.1 Ein Steuerungssystem für Insulinpumpen

1.3.2 Ein Patienteninformationssystem für die psychiatrische Ambulanz

1.3.3 Eine Wetterstation in Wildnisgebieten

1.3.4 Eine digitale Lernumgebung für Schulen

Kapitel 2 - Softwareprozesse

2.1 Vorgehensmodelle

2.1.1 Das Wasserfallmodell

2.1.2 Inkrementelle Entwicklung

2.1.3 Integration und Konfiguration

2.2 Prozessaktivitäten

2.2.1 Softwarespezifikation

2.2.2 Softwareentwurf und -implementierung

2.2.3 Softwarevalidierung

2.2.4 Weiterentwicklung von Software

2.3 Umgang mit Änderungen

2.3.1 Softwareprototypen

2.3.2 Inkrementelle Auslieferung

2.4 Prozessverbesserung

Kapitel 3 - Agile Softwareentwicklung

3.1 Agile Methoden

3.2 Techniken der agilen Entwicklung

3.2.1 User-Stories

3.2.2 Refactoring

3.2.3 Test-First-Entwicklung

3.2.4 Pair Programming

3.3 Agiles Projektmanagement

3.4 Skalieren von agilen Methoden

Inhaltsverzeichnis

- 3.4.1 Praktische Probleme mit agilen Methoden
- 3.4.2 Agile und plangesteuerte Methoden
- 3.4.3 Agile Methoden für große Systeme
- 3.4.4 Organisationsübergreifende agile Methoden

Kapitel 4 - Requirements-Engineering

- 4.1 Funktionale und nichtfunktionale Anforderungen
 - 4.1.1 Funktionale Anforderungen
 - 4.1.2 Nichtfunktionale Anforderungen
- 4.2 Prozesse des Requirements-Engineerings
- 4.3 Anforderungserhebung und -analyse
 - 4.3.1 Techniken zur Anforderungserhebung
 - 4.3.2 User-Stories und Szenarios
- 4.4 Anforderungsspezifikation
 - 4.4.1 Spezifikation in natürlicher Sprache
 - 4.4.2 Strukturierte Spezifikationen
 - 4.4.3 Anwendungsfälle
 - 4.4.4 Die Gesamtsystemspezifikation
- 4.5 Validierung von Anforderungen
- 4.6 Anforderungsänderung
 - 4.6.1 Planung des Anforderungsmanagements
 - 4.6.2 Anforderungsänderungsmanagement

Kapitel 5 - Systemmodellierung

- 5.1 Kontextmodelle
- 5.2 Interaktionsmodelle
 - 5.2.1 Anwendungsfallmodellierung
 - 5.2.2 Sequenzdiagramme
- 5.3 Strukturelle Modelle
 - 5.3.1 Klassendiagramme
 - 5.3.2 Generalisierung
 - 5.3.3 Aggregation
- 5.4 Verhaltensmodelle
 - 5.4.1 Datenorientierte Modellierung
 - 5.4.2 Ereignisgesteuerte Modellierung
 - 5.4.3 Modellgetriebene Softwareentwicklung
- 5.5 Modellgetriebene Architektur

Kapitel 6 - Entwurf der Architektur

- 6.1 Architektonische Entwurfsentscheidungen
- 6.2 Architektursichten
- 6.3 Architekturmuster
 - 6.3.1 Schichtenarchitektur
 - 6.3.2 Repository-Architektur
 - 6.3.3 Client-Server-Architektur
 - 6.3.4 Pipes-und-Filter-Architektur
- 6.4 Anwendungsarchitekturen
 - 6.4.1 Transaktionsverarbeitende Systeme

Inhaltsverzeichnis

6.4.2 Informationssysteme

6.4.3 Sprachverarbeitende Systeme

Kapitel 7 - Entwurf und Implementierung

7.1 Objektorientierter Entwurf mit UML

7.1.1 Systemkontext und Interaktionen

7.1.2 Entwurf der Architektur

7.1.3 Bestimmung der Objektklassen

7.1.4 Entwurfsmodelle

7.1.5 Schnittstellenspezifikation

7.2 Entwurfsmuster

7.3 Implementierungsaspekte

7.3.1 Wiederverwendung

7.3.2 Konfigurationsverwaltung

7.3.3 Host-Ziel-Entwicklung

7.4 Open-Source-Entwicklung

7.4.1 Open-Source-Lizenzierung

Kapitel 8 - Testen von Software

8.1 Entwicklertests

8.1.1 Modultests

8.1.2 Auswahl der Testfälle für Modultests

8.1.3 Testen von Komponenten

8.1.4 Testen von Systemen

8.2 Testgetriebene Entwicklung

8.3 Freigabetests

8.3.1 Anforderungsbasiertes Testen

8.3.2 Szenariobasiertes Testen

8.3.3 Leistungstests

8.4 Benutzertests

Kapitel 9 - Softwareevolution

9.1 Evolutionsprozesse

9.2 Altsysteme

9.2.1 Verwaltung von Altsystemen

9.3 Softwarewartung

9.3.1 Vorhersagen des Wartungsaufwands

9.3.2 Software-Reengineering

9.3.3 Refactoring

Teil II - Systemverlässlichkeit und Informationssicherheit

Kapitel 10 - Verlässliche Systeme

10.1 Eigenschaften der Verlässlichkeit

10.2 Soziotechnische Systeme

10.2.1 Regulierung und Einhaltung der Vorschriften

10.3 Redundanz und Diversität

10.4 Verlässliche Prozesse

10.5 Formale Methoden und Verlässlichkeit

Inhaltsverzeichnis

Kapitel 11 - Reliability-Engineering

- 11.1 Verfügbarkeit und Zuverlässigkeit
- 11.2 Zuverlässigkeitsanforderungen
 - 11.2.1 Zuverlässigkeitsmetriken
 - 11.2.2 Nichtfunktionale Zuverlässigkeitsanforderungen
 - 11.2.3 Funktionale Zuverlässigkeitsanforderungen
- 11.3 Fehlertolerante Architekturen
 - 11.3.1 Schutzsysteme
 - 11.3.2 Selbstüberwachende Architekturen
 - 11.3.3 Diversitäre Programmierung
 - 11.3.4 Softwarediversität
- 11.4 Zuverlässige Programmierung
- 11.5 Zuverlässigkeitsmessung
 - 11.5.1 Betriebsprofile

Kapitel 12 - Safety-Engineering

- 12.1 Betriebssicherheitskritische Systeme
- 12.2 Spezifikation der Betriebssicherheit
 - 12.2.1 Gefahrenerkennung
 - 12.2.2 Gefahrenbewertung
 - 12.2.3 Gefahrenanalyse
 - 12.2.4 Risikoreduzierung
- 12.3 Prozesse zur Entwicklung betriebssicherer Systeme
 - 12.3.1 Prozesse für die Gewährleistung der Betriebssicherheit
 - 12.3.2 Formale Verifikation
 - 12.3.3 Modellprüfung
 - 12.3.4 Statische Programmanalyse
- 12.4 Nachweis der Betriebssicherheit
 - 12.4.1 Strukturierte Argumentationen
 - 12.4.2 Betriebssicherheitsargumentationen für Software

Kapitel 13 - Security-Engineering

- 13.1 Informationssicherheit und Verlässlichkeit
- 13.2 Informationssicherheit in Unternehmen und Organisationen
 - 13.2.1 Bewertung von Informationssicherheitsrisiken
- 13.3 Anforderungen an die Informationssicherheit
 - 13.3.1 Missbrauchsfälle
- 13.4 Entwerfen sicherer Systeme
 - 13.4.1 Bewertung der Entwurfsrisiken
 - 13.4.2 Architekturentwurf
 - 13.4.3 Entwurfsrichtlinien
 - 13.4.4 Informationssichere Systemprogrammierung
- 13.5 Testen und Gewährleistung der Informationssicherheit

Kapitel 14 - Resilienz-Engineering

- 14.1 Cybersicherheit
- 14.2 Soziotechnische Resilienz

Inhaltsverzeichnis

14.2.1 Menschliches Versagen

14.2.2 Betriebs- und Managementprozesse

14.3 Entwerfen resilienter Systeme

Teil III - Software-Engineering für Fortgeschrittene

Kapitel 15 - Wiederverwendung von Software

15.1 Die Wiederverwendungslandschaft

15.2 Anwendungsframeworks

15.3 Softwareproduktlinien

15.4 Wiederverwendung von Anwendungssystemen

15.4.1 Konfigurierbare Anwendungssysteme

15.4.2 Integrierte Anwendungssysteme

Kapitel 16 - Komponentenbasiertes Software-Engineering

16.1 Komponenten und Komponentenmodelle

16.1.1 Komponentenmodelle

16.2 Prozesse des komponentenbasierten Software- Engineerings

16.2.1 Komponentenbasiertes Software-Engineering für Wiederverwendung

16.2.2 Komponentenbasiertes Software-Engineering mit Wiederverwendung

16.3 Komposition von Komponenten

Kapitel 17 - Entwicklung verteilter Systeme

17.1 Verteilte Systeme

17.1.1 Kommunikationsmodelle

17.1.2 Middleware

17.2 Client-Server-Systeme

17.3 Architekturmuster für verteilte Systeme

17.3.1 Master-Slave-Architekturen

17.3.2 Zweischichtige Client-Server-Architekturen

17.3.3 Mehrschichtige Client-Server-Architekturen

17.3.4 Verteilte Komponentenarchitekturen

17.3.5 Peer-to-Peer-Architekturen

17.4 Software als Service

Kapitel 18 - Serviceorientiertes Software- Engineering

18.1 Serviceorientierte Architektur

18.1.1 Servicekomponenten in einer serviceorientierten Architektur

18.2 REST-konforme Services

18.3 Service-Engineering

18.3.1 Ermittlung von Servicekandidaten

18.3.2 Entwerfen von Serviceschnittstellen

18.3.3 Implementierung und Bereitstellung der Services

18.4 Kombinationen von Services

18.4.1 Entwurf und Implementierung des Workflows

18.4.2 Testen von Services

Kapitel 19 - Systems-Engineering

19.1 Soziotechnische Systeme

Inhaltsverzeichnis

19.1.1 Typische Systemeigenschaften

19.1.2 Nichtdeterminismus

19.1.3 Erfolgskriterien

19.2 Entwurfsplanung

19.3 Systembeschaffung

19.4 Systementwicklung

19.5 Systembetrieb und Systemevolution

19.5.1 Systemevolution

Kapitel 20 - Systeme von Systemen

20.1 Systemkomplexität

20.2 Klassifikation von Systemen von Systemen

20.3 Reduktionismus und komplexe Systeme

20.4 SvS-Engineering

20.4.1 Schnittstellenentwicklung

20.4.2 Integration und Inbetriebnahme

20.5 Architektur von Systemen von Systemen

20.5.1 Architekturmuster für Systeme von Systemen

Kapitel 21 - Entwicklung von Echtzeitsoftware

21.1 Entwurf eingebetteter Systeme

21.1.1 Modellierung von Echtzeitsystemen

21.1.2 Programmierung von Echtzeitsystemen

21.2 Architekturmuster für Echtzeitsoftware

21.2.1 Beobachten und Reagieren

21.2.2 Umgebungssteuerung

21.2.3 Pipelineverarbeitung

21.3 Analyse des Zeitverhaltens

21.4 Echtzeitbetriebssysteme

21.4.1 Prozessverwaltung

Teil IV - Softwaremanagement

Kapitel 22 - Projektmanagement

22.1 Risikomanagement

22.1.1 Risikoerkennung

22.1.2 Risikoanalyse

22.1.3 Risikoplanung

22.1.4 Risikoüberwachung

22.2 Personalmanagement

22.2.1 Mitarbeitermotivation

22.3 Teamwork

22.3.1 Teammitglieder auswählen

22.3.2 Organisation der Gruppe

22.3.3 Kommunikation in der Gruppe

Kapitel 23 - Projektplanung

23.1 Preiskalkulation für Software

23.2 Plangesteuerte Entwicklung

Inhaltsverzeichnis

23.2.1 Projektpläne

23.2.2 Der Planungsprozess

23.3 Zeitplanung

23.3.1 Darstellung des Zeitplans

23.4 Agile Planung

23.5 Schätztechniken

23.5.1 Algorithmische Kostenmodellierung

23.6 Das COCOMO-II-Modell

23.6.1 Das Application-Composition-Modell

23.6.2 Das Early-Design-Modell

23.6.3 Das Reuse-Modell

23.6.4 Das Post-Architecture-Modell

23.6.5 Projektdauer und Personalplanung

Kapitel 24 - Qualitätsmanagement

24.1 Softwarequalität

24.2 Softwarestandards

24.2.1 Der Rahmenstandard ISO 9001

24.3 Reviews und Inspektionen

24.3.1 Der Review-Prozess

24.3.2 Programminspektionen

24.4 Qualitätsmanagement und agile Entwicklung

24.5 Softwaremessung

24.5.1 Produktmetriken

24.5.2 Softwarekomponentenanalyse

24.5.3 Mehrdeutigkeit von Messungen

24.5.4 Softwareanalytik

Kapitel 25 - Konfigurationsmanagement

25.1 Versionsmanagement

25.2 Systemerstellung

25.3 Änderungsmanagement

25.4 Release-Management

Glossar

Literatur A-Z

Register

Copyright

Copyright

Daten, Texte, Design und Grafiken dieses eBooks, sowie die eventuell angebotenen eBook-Zusatzdaten sind urheberrechtlich geschützt. Dieses eBook stellen wir lediglich als **persönliche Einzelplatz-Lizenz** zur Verfügung!

Jede andere Verwendung dieses eBooks oder zugehöriger Materialien und Informationen, einschließlich

- der Reproduktion,
- der Weitergabe,
- des Weitervertriebs,
- der Platzierung im Internet, in Intranets, in Extranets,
- der Veränderung,
- des Weiterverkaufs und
- der Veröffentlichung

bedarf der **schriftlichen Genehmigung** des Verlags. Insbesondere ist die Entfernung oder Änderung des vom Verlag vergebenen Passwort- und DRM-Schutzes ausdrücklich untersagt!

Bei Fragen zu diesem Thema wenden Sie sich bitte an: **info@pearson.de**

Zusatzdaten

Möglicherweise liegt dem gedruckten Buch eine CD-ROM mit Zusatzdaten oder ein Zugangscode zu einer eLearning Plattform bei. Die Zurverfügungstellung dieser Daten auf unseren Websites ist eine freiwillige Leistung des Verlags. **Der Rechtsweg ist ausgeschlossen.** Zugangscodes können Sie darüberhinaus auf unserer Website käuflich erwerben.

Hinweis

Dieses und viele weitere eBooks können Sie rund um die Uhr und legal auf unserer Website herunterladen:

<https://www.pearson-studium.de>