



Thomas Grechenig  
Mario Bernhart  
Roland Breiteneder  
Karin Kappel

# Softwaretechnik

Mit Fallbeispielen aus realen  
Entwicklungsprojekten

## Softwaretechnik

# Softwaretechnik

## Inhaltsverzeichnis

Softwaretechnik - Mit Fallbeispielen aus realen  
Entwicklungsprojekten

### Inhaltsverzeichnis

1 Einleitung

2 Ingenieurdisziplin Softwaretechnik und die Bausteine eines  
Projektes

3 Fallbeispiele von Projekten

4 Phase Analyse Was wird gebaut?

5 Phase Entwurf Wie wird technisch gebaut?

6 Phase Implementierung Wie wird codiert?

7 Phase Integration/Test Wie wird zusammengefügt/geprüft

8 Projektphasen: Inbetriebnahme, Rollout und Wartung

9 Vorgehensmodelle im Software Engineering

10 Projektmanagement und Risikomanagement

11 Qualitätssicherung und Qualitätsmanagement

12 Usability Engineering in der Softwareentwicklung

13 Sicherheit in der Softwareentwicklung

14 Exemplarische Vertiefungen

Literatur

Register

### 1 Einleitung

1.1 Ziel dieses Buches: Moderne Software mit Qualität.

1.2 Aufbau des Buches und Lesepfade.

1.3 Werkentstehung und Companion Website.



# **Inhaltsverzeichnis**

## **2 Ingenieurdisziplin Softwaretechnik und die Bausteine eines Projektes**

- 2.1 Wie reif ist die Softwaretechnik als Ingenieurdisziplin?
- 2.2 Industrielle Konsequenzen aus dem Reifegrad
- 2.3 Softwaretechnik als Verfahren der Übersetzung,  
Entscheidung und Schlichtung zwischen unterschiedlichen  
Denkweisen
- 2.4 Definition eines (Software-)Projektes
- 2.5 Frequently Asked Questions (FAQ) für Novizen
- 2.6 Die charakteristischen Merkmale von Softwareprojekten
- 2.7 Software = Peopleware

## **3 Fallbeispiele von Projekten**

### **3.1 LiSaMe.**

- 3.1.1 Organisatorische Aspekte
- 3.1.2 Aspekte der technischen Konzeption

### **3.2 Stadt21**

- 3.2.1 Das Projekt Stadt21
- 3.2.2 Allgemeiner Aufbau
- 3.2.3 Prozesserstellung
- 3.2.4 Aktenbearbeitung
- 3.2.5 Administration
- 3.2.6 Entwicklungsumgebung

### **3.3 Touch&Go.**

- 3.3.1 Technische Anforderungen
- 3.3.2 Architektur des Touch&Go Services
- 3.3.3 Integration unterschiedlicher Technologien
- 3.3.4 Schnittstellen und Integration mit Bestandsystemen
- 3.3.5 Organisatorische Aspekte
- 3.3.6 Entwicklungsumgebung

# Inhaltsverzeichnis

## 3.4 HISS

- 3.4.1 Projektentstehung
- 3.4.2 Ziele und Umfang von HISS
- 3.4.3 Phasen und grober Aufbau
- 3.4.4 Motivation und Ziele des Fachkonzeptes
- 3.4.5 Vorgehensweise bei der fachlichen Analyse
- 3.4.6 Die Softwareentwicklung von HISS

## 4 Phase Analyse Was wird gebaut?

### 4.1 Grundlagen der Analyse

- 4.1.1 Vision und Scope
- 4.1.2 Stakeholder und Anforderungsquellen
- 4.1.3 Machbarkeit und Machbarkeitsstudie
- 4.1.4 Make or Buy
- 4.1.5 Grobarchitektur
- 4.1.6 Anforderungsvolatilität
- 4.1.7 Klassifikation von Anforderungen
- 4.1.8 Was sind gute Anforderungen?

### 4.2 Anforderungsmodellierung

- 4.2.1 Unified Modeling Language (UML)
- 4.2.2 DSLs
- 4.2.3 Datenmodellierung

### 4.3 Anforderungsanalyse

- 4.3.1 Anwendungsfälle (Use Cases)
- 4.3.2 User Stories/Kunde vor Ort
- 4.3.3 Geschäftsprozessmodellierung
- 4.3.4 Entwurf und Prototyping der Anwenderschnittstelle
- 4.3.5 End User Programming

### 4.4 Anforderungsdokumentation.

- 4.4.1 Projektglossar
- 4.4.2 Anforderungsverwaltung

# Inhaltsverzeichnis

4.4.3 Nachverfolgbarkeit von Anforderungen (Traceability)

## 4.5 Organisatorische Aspekte der Analyse.

4.5.1 Die Rolle des Analytikers

4.5.2 Stakeholder identifizieren

4.5.3 Ziele definieren

4.5.4 Anforderungen ermitteln

## 4.6 Glossar.

# 5 Phase Entwurf Wie wird technisch gebaut?

## 5.1 Grundlagen der Softwarearchitekturen

5.1.1 Definitionen zu unterschiedlichen Begriffen

5.1.2 Von der Anforderung zur Architektur

5.1.3 Softwarearchitektur im Wandel der Zeit

5.1.4 Grundlegende Architekturmuster

5.1.5 Architekturbausteine und deren Kommunikation

5.1.6 Architektur-Frameworks

## 5.2 Entwurfsparadigmen

5.2.1 Objektorientiertes Design

5.2.2 Serviceorientiertes Design

## 5.3 Architekturmuster

5.3.1 Architektur-Patterns

5.3.2 Design-Patterns

## 5.4 State-of-Art Softwarearchitektur: serviceorientierte Architektur und Design

5.4.1 SOA-Referenzarchitekturen

5.4.2 Granularität

5.4.3 Service-Design-Paradigmen und Benefits

5.4.4 Service-Modellierung

## 5.5 Glossar

# 6 Phase Implementierung Wie wird codiert?

# Inhaltsverzeichnis

## 6.1 Grundlagen der Implementierung

### 6.1.1 Die Wahl der Programmiersprache

## 6.2 Frameworks als technische Basis der Implementierung

### 6.2.1 Warum werden Frameworks eingesetzt?

### 6.2.2 Abgrenzungen zwischen Framework und Bibliothek

### 6.2.3 Designentscheidung: Welches Framework soll eingesetzt werden?

### 6.2.4 Eigenschaften von Frameworks

## 6.3 Die Entwicklungsumgebung (IDE) als Cockpit des Entwicklers

## 6.4 Konzepte für eine wartungsfreundliche Implementierung

### 6.4.1 Best Practices für die Strukturierung des Quellcodes

### 6.4.2 Kollektiver Codebesitz (Collective Code Ownership)

### 6.4.3 Automatisierte Entwicklungstests

### 6.4.4 Refaktorisierung

### 6.4.5 Dokumentation der Entwicklungsaufträge

### 6.4.6 Implementierungsrichtlinien

### 6.4.7 Internationalisierung in der Softwareentwicklung

### 6.4.8 Fehlerbehandlung (Exception Management)

### 6.4.9 Laufzeitprotokollierung (Logging)

## 6.5 Konfigurationsmanagement in der Implementierung

### 6.5.1 Build Management

### 6.5.2 Source Code-Management

### 6.5.3 Vergabe von Programmversionennummern

## 6.6 Glossar

# 7 Phase Integration/Test Wie wird zusammengefügt/geprüft

## 7.1 Bedeutung von Softwaretests

## 7.2 Grundlagen der Integration und des Tests

### 7.2.1 Softwaretests Begriffserklärung

### 7.2.2 Ziele und Grundsätze des Softwaretests

## 7.3 Test- und Integrationsstufen

# **Inhaltsverzeichnis**

7.3.1 Komponententest

7.3.2 Integrationstest

7.3.3 Systemtest

7.3.4 Akzeptanztest

## **7.4 Funktionale Softwaretests**

7.4.1 Strukturelle Methoden

7.4.2 Funktionale Methoden

## **7.5 Nichtfunktionale Softwaretests**

7.5.1 Performance-Tests

## **7.6 Testautomatisierung**

7.6.1 Automatisierte Komponententests

7.6.2 Automatisierte GUI-Tests

7.6.3 Regressionstest

7.6.4 Testdaten

## **7.7 Testprozess**

7.7.1 Fehlermanagement

7.7.2 Software Test Outsourcing

7.7.3 Standards und Normen im Softwaretest

## **7.8 Glossar**

# **8 Projektphasen: Inbetriebnahme, Rollout und Wartung**

## **8.1 Integration der Systemanwender**

8.1.1 Anforderungen an die Anwenderdokumentation

8.1.2 Planung und Durchführung von Schulungsmaßnahmen

8.1.3 Szenario: Applikations-Rollout

## **8.2 Inbetriebsetzung und laufender Betrieb**

8.2.1 Planung, Installation und Integration

8.2.2 Abnahme komplexer Systeme

8.2.3 Überwachung und Optimierung

8.2.4 Checkliste: Betrieb light

## **8.3 Migration von existierenden Systemteilen und Daten**

# **Inhaltsverzeichnis**

- 8.3.1 Migration vs. Integration
- 8.3.2 Migrations- und Integrationsstrategien

## **8.4 Softwarewartung.**

- 8.4.1 Wartungstypen
- 8.4.2 Wartungsprozesse
- 8.4.3 Softwarewartung als Investitionsschutz
- 8.4.4 Maßnahmen während der Wartung

## **8.5 Glossar**

# **9 Vorgehensmodelle im Software Engineering**

## **9.1 Grundlagen und Bedeutung von Vorgehensmodellen**

- 9.1.1 Die Entstehungsgeschichte der Prozessmodelle

## **9.2 Fundamentale Prozessmodelle**

- 9.2.1 Wasserfallmodell
- 9.2.2 V-Modell
- 9.2.3 Spiralmodell

## **9.3 Industriell geprägte Prozessmodelle**

- 9.3.1 Rational Unified Process (RUP)
- 9.3.2 Microsoft Solutions Framework (MSF)

## **9.4 Von Personen geprägte Prozessmodelle**

- 9.4.1 Scrum
- 9.4.2 eXtreme Programming (XP)

## **9.5 Von Gremien/Konsortien geprägte Prozessmodelle**

- 9.5.1 V-Modell 97
- 9.5.2 V-Modell XT
- 9.5.3 ISO/IEC 12207

## **9.6 Diskussion: agile versus traditionelle Vorgehensmodelle**

## **9.7 Glossar**

# **10 Projektmanagement und Risikomanagement**

## **10.1 Schätzung von Entwicklungsaufwänden**

# Inhaltsverzeichnis

10.1.1 Einflussfaktoren

10.1.2 Empirische Verfahren

10.1.3 Komparative Verfahren

10.1.4 Algorithmische Verfahren

10.1.5 Schätzproblematik

## 10.2 Planung von Ergebnissen und Projektaktivitäten

10.2.1 Ablauf und Bestandteile der Projektplanung

10.2.2 Fortschreibung der Projektplanung

## 10.3 Berichtswesen

## 10.4 Projekt-Controlling

10.4.1 Qualitätskontrolle

10.4.2 Terminkontrolle

10.4.3 Kostenkontrolle

10.4.4 Risikokontrolle

10.4.5 Earned Value Management

10.4.6 Controlling-Arten

10.4.7 Typische Probleme beim Projekt-Controlling

## 10.5 Risikomanagement

10.5.1 Was ist Risikomanagement?

10.5.2 Typische Risiken in der Softwareentwicklung

## 10.6 Glossar

# 11 Qualitätssicherung und Qualitätsmanagement

## 11.1 Grundlagen der Softwarequalitätssicherung

11.1.1 Softwarequalitätsfaktoren

11.1.2 Softwarefehler, -fehlerfall und -fehlverhalten

11.1.3 Klassifikation der Ursachen von Softwarefehlern

11.1.4 Fehlerentstehung und Fehlerentdeckung

11.1.5 Kosten der Qualitätssicherung

## 11.2 Statische Qualitätssicherung

11.2.1 Statische Analyse



# **Inhaltsverzeichnis**

11.2.2 Softwaremetriken

11.2.3 Reviews

## **11.3 Dynamische Qualitätssicherung**

11.3.1 Testen

11.3.2 Dynamische Analyse

## **11.4 Organisatorische Qualitätssicherung**

11.4.1 Wissensmanagement

11.4.2 Konfigurationsmanagement

11.4.3 Templates und Checklisten

## **11.5 Qualitätsmanagementstandards**

11.5.1 ISO 9001 und ISO 9000-3

11.5.2 Capability Maturity Models CMM and CMMI

11.5.3 SPICE ISO/IEC 15504

## **11.6 Glossar**

# **12 Usability Engineering in der Softwareentwicklung**

## **12.1 Usability Engineering als zentraler Bestandteil des Softwareentwicklungsprozesses**

12.1.1 Usability Engineering

12.1.2 User-Centered Designprozess

12.1.3 Erfolgsfaktor Usability Engineering

## **12.2 Anwendbare Methoden im User-Centered Design**

12.2.1 Anforderungen

12.2.2 User Interface und Interaktionsdesign

12.2.3 Prototyping

12.2.4 Evaluierung

## **12.3 Web Usability**

12.3.1 Ausgewählte Gestaltungshinweise für Web Usability

12.3.2 Accessibility

## **12.4 Usability Engineering im realen Projekt**

12.4.1 Stadt21

# **Inhaltsverzeichnis**

12.4.2 HISS

12.5 Glossar

## **13 Sicherheit in der Softwareentwicklung**

13.1 Grundlagen der IT-Sicherheit

13.2 Vorgehensmodelle zur Entwicklung sicherer Software

13.3 Sicherheitsanforderungen in der Analyse

13.3.1 Erfassung Sicherheitsanforderungen

13.3.2 Bedrohungs- und Risikoanalyse

13.3.3 Sicherheit als Prozess

13.4 Sicherheitsaspekte beim Entwurf

13.4.1 Designprinzipien für sichere Software

13.4.2 Sicherheitsmuster

13.4.3 Angriffsmuster

13.4.4 Modellierung mittels UML

13.5 Sichere Implementierung.

13.5.1 Programmierrichtlinien

13.5.2 Sicherheitskonzepte von Programmiersprachen

13.5.3 Ausgewählte Angriffsmöglichkeiten

13.6 Sicherheitstest

13.6.1 Sicherheitsfunktionalität vs. sichere Funktionalität

13.6.2 Sicherheitstests im Softwarelebenszyklus

13.6.3 Analyse des Testobjekts

13.6.4 Testtechniken zur Überprüfung von Sicherheitsfehlern

13.7 IT-Sicherheit bei Betriebsübergabe und Betrieb.

13.8 Standards, Normen und Leitfäden.

13.9 Glossar

## **14 Exemplarische Vertiefungen**

14.1 Mobile Usability

14.1.1 Zielgruppe

# **Inhaltsverzeichnis**

14.1.2 Informationsdarstellung

14.1.3 Interaktionsformen

14.1.4 Kontext

14.1.5 Aktuelle Entwicklungen

14.2 Cloud Computing

14.3 Scientific Computing

14.4 IT-Sicherheit für Software im VoIP-Umfeld

14.4.1 Sicherheitsanforderungen an VoIP

14.4.2 Beispiele für Bedrohungen der IT-Sicherheit bei VoIP

14.4.3 Kernpunkte der Sicherheit bei VoIP

## **Literatur**

Kapitel 2

Kapitel 3

Kapitel 4

Kapitel 5

Kapitel 6

Kapitel 7

Kapitel 8

Kapitel 9

Kapitel 10

Kapitel 11

Kapitel 12

Kapitel 13

Kapitel 14

## **Register**

A

B

C

## **Inhaltsverzeichnis**

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Z

Ins Internet: Weitere Infos zum Buch, Downloads, etc.

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 Passwortschutzes ausdrücklich untersagt!

Bei Fragen zu diesem Thema wenden Sie sich bitte an: [info@pearson.de](mailto:info@pearson.de)

## Zusatzdaten

Möglicherweise liegt dem gedruckten Buch eine CD-ROM mit Zusatzdaten bei. Die Zurverfügungstellung dieser Daten auf unseren Websites ist eine freiwillige Leistung des Verlags. Der Rechtsweg ist ausgeschlossen.

## Hinweis

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



herunterladen