

Software Engineering, 8th Edition

Contents

Part 1 Introduction

1. Introduction

- 1.1 FAQs about software engineering
- 1.2 Professional and ethical responsibility

2. Socio-technical Systems

- 2.1 Emergent system properties
- 2.2 Systems engineering
- 2.3 Organizations, people and computer systems
- 2.4 Legacy systems

3. Critical Systems

- 3.1 A simple safety-critical system
- 3.2 System dependability
- 3.3 Availability and reliability
- 3.4 Safety
- 3.5 Security

4. Software Processes

- 4.1 Software process models
- 4.2 Process iteration
- 4.3 Process activities
- 4.4 The Rational Unified Process
- 4.5 Computer-aided software engineering

5. Project Management

- 5.1 Management activities
- 5.2 Project planning
- 5.3 Project scheduling
- 5.4 Risk management

Part 2: Requirements engineering

6. Software Requirements

- 6.1 Functional and non-functional requirements
- 6.2 User requirements
- 6.3 System requirements
- 6.4 Interface specification
- 6.5 The software requirements document

7. Requirements Engineering Processes

- 7.1 Feasibility studies
- 7.2 Requirements elicitation and analysis
- 7.3 Requirements validation
- 7.4 Requirements management

8. System Models

- 8.1 Context models
- 8.2 Behavioural models
- 8.3 Data models
- 8.4 Object models
- 8.5 Structured methods

9. Critical Systems Specification

- 9.1 Risk-driven specification
- 9.2 Safety specification
- 9.3 Security specification
- 9.4 Software reliability specification

10. Formal Specification

- 10.1 Formal specification in the software process
- 10.2 Sub-system interface specification
- 10.3 Behavioural specification

Part 3: Design

11. Architectural Design

- 11.1 Architectural design decision
- 11.2 System organisation
- 11.3 Decomposition styles
- 11.4 Control styles
- 11.5 Reference architectures

- 12. Distributed Systems Architecture**
 - 12.1 Multiprocessor architectures
 - 12.2 Client-server architectures
 - 12.3 Distributed object architectures
 - 12.4 Inter-organisational distributed computing

- 13. Application Architectures**
 - 13.1 Data processing systems
 - 13.2 Transaction processing systems
 - 13.3 Event processing systems
 - 13.4 Language processing systems

- 14. Object-oriented Design**
 - 14.1 Objects and object classes
 - 14.2 An object-oriented design process
 - 14.3 Design evolution

- 15. Real-time Systems**
 - 15.1 System design
 - 15.2 Real-time operating systems
 - 15.3 Monitoring and control systems
 - 15.4 Data acquisition systems

- 16. User Interface Design**
 - 16.1 Design issues
 - 16.2 The user interface design process
 - 16.3 User analysis
 - 16.4 User interface prototyping
 - 16.5 Interface evaluation

Part 4 Software development

- 17. Rapid Software Development**
 - 17.1 Agile methods
 - 17.2 Extreme programming
 - 17.3 Rapid application development
 - 17.4 Software prototyping

- 18. Software Reuse**
 - 18.1 The reuse landscape
 - 18.2 Design patterns
 - 18.3 Generator-based reuse

- 18.4 Application frameworks
- 18.5 Application system reuse

19. Component-based Software Engineering

- 19.1 Components and component models
- 19.2 The CBSE process
- 19.3 Component composition

20. Critical Systems Development

- 20.1 Dependable processes
- 20.2 Dependable programming
- 20.3 Fault tolerance
- 20.4 Fault-tolerant architectures

21. Software Evolution

- 21.1 Program evolution dynamics
- 21.2 Software maintenance
- 21.3 Evolution processes
- 21.4 Legacy system evolution

Part 5 Verification and validation

22. Verification and Validation

- 22.1 Planning verification and validation
- 22.2 Software inspections
- 22.3 Automated static analysis
- 22.4 Verification and formal methods

23. Software Testing

- 23.1 System testing
- 23.2 Component testing
- 23.3 Test case design
- 23.4 Test automation

24. Critical Systems Validation

- 24.1 Reliability validation
- 24.2 Safety assurance
- 24.3 Security assessment
- 24.4 Safety and dependability cases

Part 6 Management

25. Managing People

- 25.1 Selecting staff
- 25.2 Motivating people
- 25.3 Managing groups
- 25.4 The people capability maturity model

26. Software Cost Estimation

- 26.1 Software productivity
- 26.2 Estimation techniques
- 26.3 Algorithmic cost modelling
- 26.4 Project duration and staffing

27. Quality Management

- 27.1 Process and product quality
- 27.2 Quality assurance and standards
- 27.3 Quality planning
- 27.4 Quality control
- 27.5 Software measurement and metrics

28. Process Improvement

- 28.1 Process and product quality improvement
- 28.2 Process classification
- 28.3 Process measurement
- 28.4 Process analysis and modelling
- 28.5 Process change
- 28.6 The CMMI process improvement framework

29. Configuration Management

- 29.1 Configuration management planning
- 29.2 Change management
- 29.3 Version and release management
- 29.4 System building
- 29.5 CASE tools for configuration management

Part 7 Emerging Technologies

30. Security Engineering

- 30.1 Security concepts
- 30.2 Security risk management
- 30.3 Design for security
- 30.4 System survivability

- 31. Service-oriented software engineering**
 - 31.1 Services as reusable components
 - 31.2 Service engineering
 - 31.3 Software development with services

- 32. Aspect-oriented software development**
 - 32.1 The separation of concerns
 - 32.2 Aspects, join points and pointcuts
 - 32.3 Software engineering with aspects