

# Contents

## **Part 1 Overview**

### **Chapter 1 Introduction**

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

### **Chapter 2 Computer-based System Engineering**

- 2.1 Emergent system properties
- 2.2 Systems and their environment
- 2.3 System modelling
- 2.4 The system engineering process
- 2.5 System procurement

### **Chapter 3 Software Processes**

- 3.1 Software process models
- 3.2 Process iteration
- 3.3 Software specification
- 3.4 Software design and implementation
- 3.5 Software validation
- 3.6 Software evolution
- 3.7 Automated process support

### **Chapter 4 Project Management**

- 4.1 Management activities
- 4.2 Project planning
- 4.3 Project scheduling
- 4.4 Risk management

## **Part 2 Requirements**

### **Chapter 5 Software Requirements**

- 5.1 Functional and non-functional requirements
- 5.2 User requirements
- 5.3 System requirements
- 5.4 The software requirements document

### **Chapter 6 Requirements Engineering Processes**

- 6.1 Feasibility studies
- 6.2 Requirements elicitation and analysis
- 6.3 Requirements validation
- 6.4 Requirements management

### **Chapter 7 System Models**

- 7.1 Context models
- 7.2 Behavioural models
- 7.3 Data models
- 7.4 Object models
- 7.5 CASE workbenches

<b>Chapter 8</b>	<b>Software Prototyping</b>
8.1	Prototyping in the software process
8.2	Rapid prototyping techniques
8.3	User interface prototyping
<b>Chapter 9</b>	<b>Formal Specification</b>
9.1	Formal specification in the software process
9.2	Interface specification
9.3	Behavioural specification
<b>Part 3</b>	<b>Design</b>
<b>Chapter 10</b>	<b>Architectural Design</b>
10.1	System structuring
10.2	Control models
10.3	Modular decomposition
10.4	Domain-specific architectures
<b>Chapter 11</b>	<b>Distributed Systems Architectures</b>
11.1	Multiprocessor architectures
11.2	Client-server architectures
11.3	Distributed object architectures
11.4	CORBA
<b>Chapter 12</b>	<b>Object-oriented Design</b>
12.1	Objects and object classes
12.2	An object-oriented design process
12.3	Design evolution
<b>Chapter 13</b>	<b>Real-time Software Design</b>
13.1	System design
13.2	Real-time executives
13.3	Monitoring and control systems
13.4	Data acquisition systems
<b>Chapter 14</b>	<b>Design with Reuse</b>
14.1	Component-based development
14.2	Application families
14.3	Design patterns
<b>Chapter 15</b>	<b>User Interface Design</b>
15.1	User interface design principles
15.2	User interaction
15.3	Information presentation
15.4	User support
15.5	Interface evaluation
<b>Part 4</b>	<b>Critical Systems Engineering</b>
<b>Chapter 16</b>	<b>Dependability</b>
16.1	Critical systems
16.2	Availability and reliability

16.3	Safety
16.4	Security
<b>Chapter 17</b>	<b>Critical Systems Specification</b>
17.1	Software reliability specification
17.2	Safety specification
17.3	Security specification
<b>Chapter 18</b>	<b>Critical Systems Development</b>
18.1	Fault minimisation
18.2	Fault tolerance
18.3	Fault tolerant architectures
18.4	Safe system design
<b>Part 5</b>	<b>Verification and Validation</b>
<b>Chapter 19</b>	<b>Verification and Validation</b>
19.1	Verification and validation planning
19.2	Software inspections
19.3	Automated static analysis
19.4	Cleanroom software development
<b>Chapter 20</b>	<b>Software Testing</b>
20.1	Defect testing
20.2	Integration testing
20.3	Object-oriented testing
20.4	Testing workbenches
<b>Chapter 21</b>	<b>Critical Systems Validation</b>
21.1	Formal methods and critical systems
21.2	Reliability validation
21.3	Safety assurance
21.4	Security assessment
<b>Part 6</b>	<b>Management</b>
<b>Chapter 22</b>	<b>Managing People</b>
22.1	Limits to thinking
22.2	Group working
22.3	Choosing and keeping people
22.4	The people capability maturity model
<b>Chapter 23</b>	<b>Software Cost Estimation</b>
23.1	Productivity
23.2	Estimation techniques
23.3	Algorithmic cost modelling
23.4	Project duration and staffing
<b>Chapter 24</b>	<b>Quality Management</b>
24.1	Quality assurance and standards
24.2	Quality planning
24.3	Quality control
24.4	Software measurement and metrics

<b>Chapter 25</b>	<b>Process Improvement</b>
25.1	Process and product quality
25.2	Process analysis and modelling
25.3	Process measurement
25.4	The SEI process capability maturity model
25.5	Process classification

## **Part 7      Evolution**

### **Chapter 26      Legacy Systems**

- 26.1      Legacy system structures
- 26.2      Legacy system design
- 26.3      Legacy system assessment

### **Chapter 27      Software Change**

- 27.1      Program evolution dynamics
- 27.2      Software maintenance
- 27.3      Architectural evolution

### **Chapter 28      Software Re-engineering**

- 28.1      Source code translation
- 28.2      Reverse engineering
- 28.3      Program structure improvement
- 28.4      Program modularisation
- 28.5      Data re-engineering

### **Chapter 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

## **References**

## **Index**