

经 典 原 版 书 库

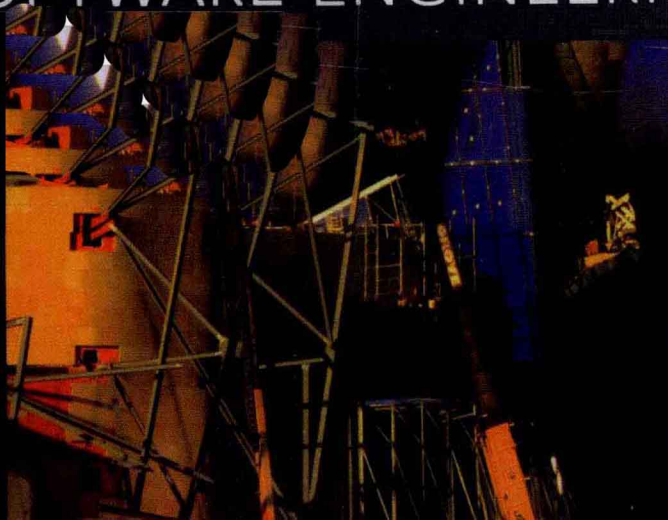
# 软件工程

(英) Ian Sommerville 著  
圣安德鲁斯大学

(英文版·第9版)

SOMMERVILLE

SOFTWARE ENGINEERING



经 典 原 版 书 库

# 软件工程

(英文版·第9版)

*Software Engineering*  
(Ninth Edition)

(英) Ian Sommerville 著  
圣安德鲁斯大学



机械工业出版社  
China Machine Press

English reprint edition copyright © 2011 by Pearson Education Asia Limited and China Machine Press.

Original English language title: *Software Engineering, Ninth Edition* (ISBN 978-0-13-703515-1) by Ian Sommerville, Copyright © 2011, 2006, 2005, 2001, 1996.

All rights reserved.

Published by arrangement with the original publisher, Pearson Education, Inc., publishing as Addison-Wesley.

For sale and distribution in the People's Republic of China exclusively (except Taiwan, Hong Kong SAR and Macau SAR).

本书英文影印版由Pearson Education Asia Ltd. 授权机械工业出版社独家出版。未经出版者书面许可, 不得以任何方式复制或抄袭本书内容。

仅限于中华人民共和国境内(不包括中国香港、澳门特别行政区和中国台湾地区)销售发行。  
本书封面贴有Pearson Education(培生教育出版集团)激光防伪标签, 无标签者不得销售。

封底无防伪标均为盗版

版权所有, 侵权必究

本书法律顾问 北京市展达律师事务所

本书版权登记号: 图字: 01-2011-2883

### 图书在版编目(CIP)数据

软件工程(英文版·第9版)/(英)萨默维尔(Sommerville, I.)著. —北京: 机械工业出版社, 2011.6

(经典原版书库)

书名原文: *Software Engineering, Ninth Edition*

ISBN 978-7-111-34825-2

I. 软… II. 萨… III. 软件工程—教材—英文 IV. TP311.5

中国版本图书馆CIP数据核字(2011)第097868号

机械工业出版社(北京市西城区百万庄大街22号 邮政编码 100037)

责任编辑: 迟振春

北京京师印务有限公司印刷

2011年6月第1版第1次印刷

170mm×242mm·49.5印张

标准书号: ISBN 978-7-111-34825-2

定价: 99.00元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88378991, 88361066

购书热线: (010) 68326294, 88379649, 68995259

投稿热线: (010) 88379604

读者信箱: [hzjsj@hzbook.com](mailto:hzjsj@hzbook.com)

## 出版者的话

文艺复兴以降，源远流长的科学精神和逐步形成的学术规范，使西方国家在自然科学的各个领域取得了垄断性的优势；也正是这样的传统，使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中，美国的产业界与教育界越来越紧密地结合，计算机学科中的许多泰山北斗同时身处科研和教学的最前线，由此而产生的经典科学著作，不仅擘划了研究的范畴，还揭示了学术的源变，既遵循学术规范，又自有学者个性，其价值并不会因年月的流逝而减退。

近年，在全球信息化大潮的推动下，我国的计算机产业发展迅猛，对专业人才的需求日益迫切。这对计算机教育界和出版界都既是机遇，也是挑战；而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短的现状下，美国等发达国家在其计算机科学发展的几十年间积淀和发展的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀计算机教材将对我国计算机教育事业的发展起到积极的推动作用，也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章公司较早意识到“出版要为教育服务”。自1998年开始，我们就将工作重点放在了遴选、移译国外优秀教材上。经过多年的不懈努力，我们与Pearson, McGraw-Hill, Elsevier, MIT, John Wiley & Sons, Cengage等世界著名出版公司建立了良好的合作关系，从他们现有的数百种教材中甄选出Andrew S. Tanenbaum, Bjarne Stroustrup, Brian W. Kernighan, Dennis Ritchie, Jim Gray, Alfred V. Aho, John E. Hopcroft, Jeffrey D. Ullman, Abraham Silberschatz, William Stallings, Donald E. Knuth, John L. Hennessy, Larry L. Peterson等大师名家的一批经典作品，以“计算机科学丛书”为总称出版，供读者学习、研究及珍藏。大理石纹理的封面，也正体现了这套丛书的品位和格调。

“计算机科学丛书”的出版工作得到了国内外学者的鼎力襄助，国内的专家不仅提供了中肯的选题指导，还不辞劳苦地担任了翻译和审校的工作；而原书的作者也相当关注其作品在中国的传播，有的还专程为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近两百个品种，这些书籍在读者中树立了良好的口碑，并被许多高校采用为正式教材和参考书籍。其影印版“经典原版书库”作为姊妹篇也被越来越多实施双语教学的学校所采用。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑，这些因素使我们的图书有了质量的保证。随着计算机科学与技术专业学科建设的不断完善和教材改革的逐渐深化，教育界对国外计算机教材的需求和应用都将步入一个新的阶段，我们的目标是尽善尽美，而反馈的意见正是我们达到这一终极目标的重要帮助。华章公司欢迎老师和读者对我们的工作提出建议或给予指正，我们的联系方式如下：

华章网站：[www.hzbook.com](http://www.hzbook.com)

电子邮件：[hzjsj@hzbook.com](mailto:hzjsj@hzbook.com)

联系电话：(010) 88379604

联系地址：北京市西城区百万庄南街1号

邮政编码：100037



华章教育

华章科技图书出版中心



# PREFACE

As I was writing the final chapters in this book in the summer of 2009, I realized that software engineering was 40 years old. The name ‘software engineering’ was proposed in 1969 at a NATO conference to discuss software development problems—large software systems were late, did not deliver the functionality needed by their users, cost more than expected, and were unreliable. I did not attend that conference but, a year later, I wrote my first program and started my professional life in software.

Progress in software engineering has been remarkable over my professional lifetime. Our societies could not function without large, professional software systems. For building business systems, there is an alphabet soup of technologies—J2EE, .NET, SaaS, SAP, BP4WS, SOAP, CBSE, etc.—that support the development and deployment of large enterprise applications. National utilities and infrastructure—energy, communications, and transport—all rely on complex and mostly reliable computer systems. Software has allowed us to explore space and to create the World Wide Web, the most significant information system in the history of mankind. Humanity is now faced with a new set of challenges—climate change and extreme weather, declining natural resources, an increasing world population to be fed and housed, international terrorism, and the need to help elderly people lead satisfying and fulfilled lives. We need new technologies to help us address these problems and, for sure, software will play a central role in these technologies.

Software engineering is, therefore, a critically important technology for the future of mankind. We must continue to educate software engineers and develop the discipline so that we can create more complex software systems. Of course, there are still problems with software projects. Software is still sometimes late and costs more than expected. However, we should not let these problems conceal the real successes in software engineering and the impressive software engineering methods and technologies that have been developed.

Software engineering is now such a huge area that it is impossible to cover the whole subject in one book. My focus, therefore, is on key topics that are fundamental

to all development processes and topics concerned with the development of reliable, distributed systems. There is an increased emphasis on agile methods and software reuse. I strongly believe that agile methods have their place but so too does ‘traditional’ plan-driven software engineering. We need to combine the best of these approaches to build better software systems.

Books inevitably reflect the opinions and prejudices of their authors. Some readers will inevitably disagree with my opinions and with my choice of material. Such disagreement is a healthy reflection of the diversity of the discipline and is essential for its evolution. Nevertheless, I hope that all software engineers and software engineering students can find something of interest here.

---

## Integration with the Web

---

There is an incredible amount of information on software engineering available on the Web and some people have questioned if textbooks like this one are still needed. However, the quality of available information is very patchy, information is sometimes presented badly and it can be hard to find the information that you need. Consequently, I believe that textbooks still have an important role to play in learning. They serve as a roadmap to the subject and allow information on method and techniques to be organized and presented in a coherent and readable way. They also provide a starting point for deeper exploration of the research literature and material available on the Web.

I strongly believe that textbooks have a future but only if they are integrated with and add value to material on the Web. This book has therefore been designed as a hybrid print/web text in which core information in the printed edition is linked to supplementary material on the Web. Almost all chapters include specially written ‘web sections’ that add to the information in that chapter. There are also four ‘web chapters’ on topics that I have not covered in the print version of the book.

The website that is associated with the book is:

**<http://www.SoftwareEngineering-9.com>**

The book’s web has four principal components:

1. *Web sections* These are extra sections that add to the content presented in each chapter. These web sections are linked from breakout boxes in each chapter.
2. *Web chapters* There are four web chapters covering formal methods, interaction design, documentation, and application architectures. I may add other chapters on new topics during the lifetime of the book.
3. *Material for instructors* The material in this section is intended to support people who are teaching software engineering. See the “Support Materials” section in this Preface.
4. *Case studies* These provide additional information about the case studies used in the book (insulin pump, mental health-care system, wilderness weather system)



as well as information about further case studies, such as the failure of the Ariane 5 launcher.

As well as these sections, there are also links to other sites with useful material on software engineering, further reading, blogs, newsletters, etc.

I welcome your constructive comments and suggestions about the book and the website. You can contact me at [ian@SoftwareEngineering-9.com](mailto:ian@SoftwareEngineering-9.com). Please include [SE9] in the subject of your message. Otherwise, my spam filters will probably reject your mail and you will not receive a reply. I do not have time to help students with their homework, so please don't ask.

## Readership

---

The book is primarily aimed at university and college students taking introductory and advanced courses in software and systems engineering. Software engineers in the industry may find the book useful as general reading and as a means of updating their knowledge on topics such as software reuse, architectural design, dependability and security, and process improvement. I assume that readers have completed an introductory programming course and are familiar with programming terminology.

## Changes from previous editions

---

This edition has retained the fundamental material on software engineering that was covered in previous editions but I have revised and updated all chapters and have included new material on many different topics. The most important changes are:

1. The move from a print-only book to a hybrid print/web book with the web material tightly integrated with the sections in the book. This has allowed me to reduce the number of chapters in the book and to focus on core material in each chapter.
2. Complete restructuring to make it easier to use the book in teaching software engineering. The book now has four rather than eight parts and each part may be used on its own or in combination with other parts as the basis of a software engineering course. The four parts are an introduction to software engineering, dependability and security, advanced software engineering, and software engineering management.
3. Several topics from previous editions are presented more concisely in a single chapter, with extra material moved onto the Web.
4. Additional web chapters, based on chapters from previous editions that I have not included here, are available on the Web.



5. I have updated and revised the content in all chapters. I estimate that between 30% and 40% of the text has been completely rewritten.
6. I have added new chapters on agile software development and embedded systems.
7. As well as these new chapters, there is new material on model-driven engineering, open source development, test-driven development, Reason's Swiss Cheese model, dependable systems architectures, static analysis and model checking, COTS reuse, software as a service, and agile planning.
8. A new case study on a patient record system for patients who are undergoing treatment for mental health problems has been used in several chapters.

## Using the book for teaching

---

I have designed the book so that it can be used in three different types of software engineering courses:

1. *General introductory courses in software engineering* The first part of the book has been designed explicitly to support a one-semester course in introductory software engineering.
2. *Introductory or intermediate courses on specific software engineering topics* You can create a range of more advanced courses using the chapters in Parts 2–4. For example, I have taught a course in critical systems engineering using the chapters in Part 2 plus chapters on quality management and configuration management.
3. *More advanced courses in specific software engineering topics* In this case, the chapters in the book form a foundation for the course. These are then supplemented with further reading that explores the topic in more detail. For example, a course on software reuse could be based around Chapters 16, 17, 18, and 19.

More information about using the book for teaching, including a comparison with previous editions, is available on the book's website.

## Support materials

---

A wide range of support material is available to help people using the book for teaching software engineering courses. This includes:

- PowerPoint presentations for all of the chapters in the book.
- Figures in PowerPoint.

- An instructor's guide that gives advice on how to use the book in different courses and explains the relationship between the chapters in this edition and previous editions.
- Further information on the book's case studies.
- Additional case studies that may be used in software engineering courses.
- Additional PowerPoint presentations on systems engineering.
- Four web chapters covering formal methods, interaction design, application architectures, and documentation.

All of this material is available free to readers of the book from the book's website or from the Pearson support site below. Additional material for instructors is available on a restricted basis to accredited instructors only:

- Model answers to selected end-of-chapter exercises.
- Quiz questions and answers for each chapter.

All support material, including restricted material, is available from:

**<http://www.pearsonhighered.com/sommerville/>**

Instructors using the book for teaching may obtain a password to access restricted material by registering at the Pearson website, by contacting their local Pearson representative, or by requesting a password by e-mail from [computing@aw.com](mailto:computing@aw.com). Passwords are not available from the author.

## Acknowledgments

---

A large number of people have contributed over the years to the evolution of this book and I'd like to thank everyone (reviewers, students, and book users) who have commented on previous editions and made constructive suggestions for change.

I'd particularly like to thank my family (Anne, Ali, and Jane) for their help and support while the book was being written. A big thank-you especially to my daughter, Jane, who discovered a talent for proofreading and editing. She was tremendously helpful in reading the entire book and did a great job spotting and fixing a large number of typos and grammatical errors.

Ian Sommerville  
October 2009



# CONTENTS

Preface	v
<b>Part 1 Introduction to Software Engineering</b>	<b>1</b>
Chapter 1 Introduction	3
1.1 Professional software development	5
1.2 Software engineering ethics	14
1.3 Case studies	17
Chapter 2 Software processes	27
2.1 Software process models	29
2.2 Process activities	36
2.3 Coping with change	43
2.4 The rational unified process	50
Chapter 3 Agile software development	56
3.1 Agile methods	58
3.2 Plan-driven and agile development	62

---

3.3	Extreme programming	64
3.4	Agile project management	72
3.5	Scaling agile methods	74
<b>Chapter 4</b>	<b>Requirements engineering</b>	<b>82</b>
4.1	Functional and non-functional requirements	84
4.2	The software requirements document	91
4.3	Requirements specification	94
4.4	Requirements engineering processes	99
4.5	Requirements elicitation and analysis	100
4.6	Requirements validation	110
4.7	Requirements management	111
<b>Chapter 5</b>	<b>System modeling</b>	<b>118</b>
5.1	Context models	121
5.2	Interaction models	124
5.3	Structural models	129
5.4	Behavioral models	133
5.5	Model-driven engineering	138
<b>Chapter 6</b>	<b>Architectural design</b>	<b>147</b>
6.1	Architectural design decisions	151
6.2	Architectural views	153
6.3	Architectural patterns	155
6.4	Application architectures	164
<b>Chapter 7</b>	<b>Design and implementation</b>	<b>176</b>
7.1	Object-oriented design using the UML	178
7.2	Design patterns	189

	7.3	Implementation issues	193
	7.4	Open source development	198
<b>Chapter 8</b>		<b>Software testing</b>	<b>205</b>
	8.1	Development testing	210
	8.2	Test-driven development	221
	8.3	Release testing	224
	8.4	User testing	228
<b>Chapter 9</b>		<b>Software evolution</b>	<b>234</b>
	9.1	Evolution processes	237
	9.2	Program evolution dynamics	240
	9.3	Software maintenance	242
	9.4	Legacy system management	252
<b>Part 2</b>		<b>Dependability and Security</b>	<b>261</b>
<b>Chapter 10</b>		<b>Sociotechnical systems</b>	<b>263</b>
	10.1	Complex systems	266
	10.2	Systems engineering	273
	10.3	System procurement	275
	10.4	System development	278
	10.5	System operation	281
<b>Chapter 11</b>		<b>Dependability and security</b>	<b>289</b>
	11.1	Dependability properties	291
	11.2	Availability and reliability	295
	11.3	Safety	299
	11.4	Security	302

<b>Chapter 12</b>	<b>Dependability and security specification</b>	<b>309</b>
	12.1 Risk-driven requirements specification	311
	12.2 Safety specification	313
	12.3 Reliability specification	320
	12.4 Security specification	329
	12.5 Formal specification	333
<b>Chapter 13</b>	<b>Dependability engineering</b>	<b>341</b>
	13.1 Redundancy and diversity	343
	13.2 Dependable processes	345
	13.3 Dependable system architectures	348
	13.4 Dependable programming	355
<b>Chapter 14</b>	<b>Security engineering</b>	<b>366</b>
	14.1 Security risk management	369
	14.2 Design for security	375
	14.3 System survivability	386
<b>Chapter 15</b>	<b>Dependability and security assurance</b>	<b>393</b>
	15.1 Static analysis	395
	15.2 Reliability testing	401
	15.3 Security testing	404
	15.4 Process assurance	406
	15.5 Safety and dependability cases	410
<b>Part 3</b>	<b>Advanced Software Engineering</b>	<b>423</b>
<b>Chapter 16</b>	<b>Software reuse</b>	<b>425</b>
	16.1 The reuse landscape	428
	16.2 Application frameworks	431

	16.3 Software product lines	434
	16.4 COTS product reuse	440
<b>Chapter 17</b>	<b>Component-based software engineering</b>	<b>452</b>
	17.1 Components and component models	455
	17.2 CBSE processes	461
	17.3 Component composition	468
<b>Chapter 18</b>	<b>Distributed software engineering</b>	<b>479</b>
	18.1 Distributed systems issues	481
	18.2 Client–server computing	488
	18.3 Architectural patterns for distributed systems	490
	18.4 Software as a service	501
<b>Chapter 19</b>	<b>Service-oriented architecture</b>	<b>508</b>
	19.1 Services as reusable components	514
	19.2 Service engineering	518
	19.3 Software development with services	527
<b>Chapter 20</b>	<b>Embedded software</b>	<b>537</b>
	20.1 Embedded systems design	540
	20.2 Architectural patterns	547
	20.3 Timing analysis	554
	20.4 Real-time operating systems	558
<b>Chapter 21</b>	<b>Aspect-oriented software engineering</b>	<b>565</b>
	21.1 The separation of concerns	567
	21.2 Aspects, join points and pointcuts	571
	21.3 Software engineering with aspects	576



<b>Part 4</b>	<b>Software Management</b>	<b>591</b>
Chapter 22	Project management	593
	22.1 Risk management	595
	22.2 Managing people	602
	22.3 Teamwork	607
Chapter 23	Project planning	618
	23.1 Software pricing	621
	23.2 Plan-driven development	623
	23.3 Project scheduling	626
	23.4 Agile planning	631
	23.5 Estimation techniques	633
Chapter 24	Quality management	651
	24.1 Software quality	655
	24.2 Software standards	657
	24.3 Reviews and inspections	663
	24.4 Software measurement and metrics	668
Chapter 25	Configuration management	681
	25.1 Change management	685
	25.2 Version management	690
	25.3 System building	693
	25.4 Release management	699
Chapter 26	Process improvement	705
	26.1 The process improvement process	708
	26.2 Process measurement	711

26.3 Process analysis	715
26.4 Process change	718
26.5 The CMMI process improvement framework	721
Glossary	733
Subject Index	749
Author Index	767