

# ENGINEERING SOFTWARE PRODUCTS

---

An Introduction to Modern Software  
Engineering

Ian Sommerville



Senior Vice President Courseware Portfolio Management: Engineering, Computer Science, Mathematics, Statistics, and Global Editions: *Marcia J. Horton*  
Vice President, Portfolio Management: Engineering, Computer Science, and Global Editions: *Julian Partridge*  
Executive Portfolio Manager: *Matt Goldstein*  
Portfolio Management Assistant: *Meghan Jacoby*  
Managing Producer, ECS and Mathematics: *Scott Disanno*  
Senior Content Producer: *Erin Ault*  
Project Manager: *Rose Kernan*  
Manager, Rights and Permissions: *Ben Ferrini*  
Operations Specialist: *Maura Zaldivar-Garcia*  
Inventory Manager: *Bruce Boundy*  
Product Marketing Manager: *Yvonne Vannatta*  
Field Marketing Manager: *Demetrius Hall*  
Marketing Assistant: *Jon Bryant*  
Cover Design: *Black Horse Designs*  
Cover Image: *Dmitrydesign/Shutterstock*  
Composition: Pearson CSC  
Cover Printer: *Phoenix Color/Hagerstown*  
Printer/Binder: *Lake Side Communications, Inc. (LSC)*

**Copyright © 2020 Pearson Education, Inc. 221 River Street, Hoboken, NJ 07030. All rights reserved.** Manufactured in the United States of America. This publication is protected by copyright, and permission should be obtained from the publisher prior to any prohibited reproduction, storage in a retrieval system, or transmission in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise. For information regarding permissions, request forms and the appropriate contacts within the Pearson Education Global Rights & Permissions department, please visit [www.pearsoned.com/permissions/](http://www.pearsoned.com/permissions/).

Many of the designations by manufacturers and seller to distinguish their products are claimed as trademarks. Where those designations appear in this book, and the publisher was aware of a trademark claim, the designations have been printed in initial caps or all caps.

The author and publisher of this book have used their best efforts in preparing this book. These efforts include the development, research, and testing of the theories and programs to determine their effectiveness. The author and publisher make no warranty of any kind, expressed or implied, with regard to these programs or the documentation contained in this book. The author and publisher shall not be liable in any event for incidental or consequential damages in connection with, or arising out of, the furnishing, performance, or use of these programs.

#### Library of Congress Cataloging-in-Publication Data

Names: Sommerville, Ian, 1951—author.  
Title: Engineering software products / Ian Sommerville.  
Description: First edition. | Hoboken, NJ : Pearson, 2019. | Includes bibliographical references and indexes.  
Identifiers: LCCN 2019004631 | ISBN 9780135210642  
Subjects: LCSH: Software engineering. | Computer software—Development.  
Classification: LCC QA76.758 .S67 2019 | DDC 005.1—dc23 LC record available at <https://lccn.loc.gov/2019004631>

1 19



ISBN 10: 0-13-521064-X  
ISBN 13: 978-0-13-521064-2

# PREFACE

---

Software products, such as stand-alone programs, web apps and services, and mobile apps, have transformed our everyday life and work. There are tens of thousands of software product companies, and hundreds of thousands of software engineers are employed worldwide in software product development.

Contrary to what some people may think, engineering software products needs more than coding skills. So, I've written this book to introduce some of the software engineering activities that are important for the production of reliable and secure software products.

## Who is the book for?

---

The book has been designed for students taking a first course in software engineering. People thinking about developing a product who don't have much software engineering experience may also find it useful.

## Why do we need a software engineering book that's focused on software products?

---

Most software engineering texts focus on *project-based* software engineering, where a client develops a specification and the software is developed by another company. However, the software engineering methods and techniques that have been developed for large-scale projects are not suited to software product development.

Students often find it difficult to relate to large, custom software systems. I think that students find it easier to understand software engineering techniques when they are relevant to the type of software that they constantly use. Also, many product engineering techniques are more directly relevant to student projects than project-oriented techniques.

## Is this a new edition of your other software engineering textbook?

---

No, this book takes a completely different approach and, apart from a couple of diagrams, does not reuse any material from *Software Engineering*, 10th edition.

## What's in the book?

---

Ten chapters cover software products, agile software engineering, features, scenarios and user stories, software architecture, cloud-based software, microservices architecture, security and privacy, reliable programming, testing, and DevOps and code management.

I've designed the book so that it's suitable for a one-semester software engineering course.

## How is this book different from other introductory texts on software engineering?

---

As I said, the focus is on *products* rather than *projects*. I cover techniques that most other SE texts don't cover, such as personas and scenarios, cloud computing, microservices, security, and DevOps. As product innovation doesn't come from university research, there are no citations or references to research and the book is written in an informal style.

---

## What do I need to know to get value from the book?

---

I assume that you have programming experience with a modern object-oriented programming language such as Java or Python and that you are familiar with good programming practice, such as the use of meaningful names. You should also understand basic computing concepts, such as objects, classes, and databases. The program examples in the book are written in Python, but they are understandable by anyone with programming experience.

## What extra material is available to help teachers and instructors?

---

1. An instructor's manual with solutions to exercises and quiz questions for all chapters
2. Suggestions how you can use the book in a one-semester software engineering course
3. Presentations for teaching (Keynote, PowerPoint, and PDF)

You can access this material at: <https://www.pearsonhighered.com/sommerville>

Additional material is available on the book's website

<https://iansommerville.com/engineering-software-products/>

## Where can I find out more?

---

I've written a couple of blog posts that are relevant to the book. These provide more information about my thoughts on teaching software engineering and my motivation for writing the book.

"Out with the UML (and other stuff too): reimagining introductory courses in software engineering"

<https://iansommerville.com/systems-software-and-technology/what-should-we-teach-in-software-engineering-courses/>

"Engineering Software Products"

<https://iansommerville.com/systems-software-and-technology/engineering-software-products/>

## Acknowledgments

---

I'd like to thank the reviewers who made helpful and supportive suggestions when they reviewed the initial proposal for this book:

Paul Eggert—*UCLA Los Angeles*  
Jeffrey Miller—*University of Southern California*  
Harvey Siy—*University of Nebraska Omaha*  
Edmund S. Yu—*Syracuse University*  
Gregory Gay—*University of South Carolina*  
Josh Delinger—*Towson University*  
Rocky Slavin—*University of Texas San Antonio*  
Bingyang Wei—*Midwestern State University*

Thanks also to Adam Barker from St. Andrews University for keeping me right on containers and to Rose Kernan who managed the production of the book.

Thanks, as ever, to my family for their help and support while I was writing the book. Particular thanks to my daughter Jane, who did a great job of reading and commenting on the text. She was a brutal editor! Her suggested changes significantly improved the quality of my prose.

Finally, special thanks to our newest family member, my beautiful grandson Cillian, who was born while I was writing this book. His bubbly personality and constant smiles were a very welcome distraction from the sometimes tedious job of book writing and editing.

*Ian Sommerville*

# CONTENTS

|                  |                                   |           |
|------------------|-----------------------------------|-----------|
| <b>Chapter 1</b> | <b>Software Products</b>          | <b>1</b>  |
| 1.1              | The product vision                | 7         |
| 1.2              | Software product management       | 11        |
| 1.3              | Product prototyping               | 16        |
|                  | Key Points                        | 17        |
|                  | Recommended Reading               | 18        |
|                  | Presentations, Videos, and Links  | 18        |
|                  | Exercises                         | 19        |
| <br>             |                                   |           |
| <b>Chapter 2</b> | <b>Agile Software Engineering</b> | <b>20</b> |
| 2.1              | Agile methods                     | 20        |
| 2.2              | Extreme Programming               | 24        |
| 2.3              | Scrum                             | 27        |
|                  | Key Points                        | 47        |
|                  | Recommended Reading               | 48        |
|                  | Presentations, Videos, and Links  | 48        |
|                  | Exercises                         | 49        |

---

|                  |   |            |
|------------------|---|------------|
| <b>Chapter 3</b> | <b>Features, Scenarios, and Stories</b> | <b>50</b>  |
| 3.1              | Personas                                | 54         |
| 3.2              | Scenarios                               | 59         |
| 3.3              | User stories                            | 66         |
| 3.4              | Feature identification                  | 70         |
|                  | Key Points                              | 79         |
|                  | Recommended Reading                     | 80         |
|                  | Presentations, Videos, and Links        | 80         |
|                  | Exercises                               | 80         |
| <br>             |   |            |
| <b>Chapter 4</b> | <b>Software Architecture</b>            | <b>82</b>  |
| 4.1              | Why is architecture important?          | 84         |
| 4.2              | Architectural design                    | 88         |
| 4.3              | System decomposition                    | 92         |
| 4.4              | Distribution architecture               | 103        |
| 4.5              | Technology issues                       | 109        |
|                  | Key Points                              | 113        |
|                  | Recommended Reading                     | 114        |
|                  | Presentations, Videos, and Links        | 114        |
|                  | Exercises                               | 115        |
| <br>             |   |            |
| <b>Chapter 5</b> | <b>Cloud-Based Software</b>             | <b>116</b> |
| 5.1              | Virtualization and containers           | 118        |
| 5.2              | Everything as a service                 | 124        |
| 5.3              | Software as a service                   | 127        |
| 5.4              | Multi-tenant and multi-instance systems | 132        |
| 5.5              | Cloud software architecture             | 140        |
|                  | Key Points                              | 147        |
|                  | Recommended Reading                     | 148        |



---

|   |            |
|---|------------|
| Presentations, Videos, and Links            | 149        |
| Exercises                                   | 149        |
| <b>Chapter 6</b> Microservices Architecture | <b>150</b> |
| 6.1 Microservices                           | 154        |
| 6.2 Microservices architecture              | 157        |
| 6.3 RESTful services                        | 173        |
| 6.4 Service deployment                      | 179        |
| Key Points                                  | 182        |
| Recommended Reading                         | 183        |
| Presentations, Videos, and Links            | 184        |
| Exercises                                   | 184        |
| <b>Chapter 7</b> Security and Privacy       | <b>185</b> |
| 7.1 Attacks and defenses                    | 188        |
| 7.2 Authentication                          | 195        |
| 7.3 Authorization                           | 201        |
| 7.4 Encryption                              | 203        |
| 7.5 Privacy                                 | 213        |
| Key Points                                  | 217        |
| Recommended Reading                         | 218        |
| Presentations, Videos, and Links            | 219        |
| Exercises                                   | 219        |
| <b>Chapter 8</b> Reliable Programming       | <b>221</b> |
| 8.1 Fault avoidance                         | 223        |
| 8.2 Input validation                        | 242        |
| 8.3 Failure management                      | 249        |
| Key Points                                  | 256        |

|  |            |
|--|------------|
| Recommended Reading                          | 256        |
| Presentations, Videos, and Links             | 257        |
| Exercises                                    | 257        |
| <b>Chapter 9 Testing</b>                     | <b>259</b> |
| 9.1 Functional testing                       | 262        |
| 9.2 Test automation                          | 273        |
| 9.3 Test-driven development                  | 281        |
| 9.4 Security testing                         | 285        |
| 9.5 Code reviews                             | 288        |
| Key Points                                   | 292        |
| Recommended Reading                          | 292        |
| Presentations, Videos, and Links             | 293        |
| Exercises                                    | 293        |
| <b>Chapter 10 DevOps and Code Management</b> | <b>295</b> |
| 10.1 Code management                         | 299        |
| 10.2 DevOps automation                       | 310        |
| 10.3 DevOps measurement                      | 321        |
| Key Points                                   | 326        |
| Recommended Reading                          | 326        |
| Presentations, Videos, and Links             | 327        |
| Exercises                                    | 327        |
| Index  | 329        |