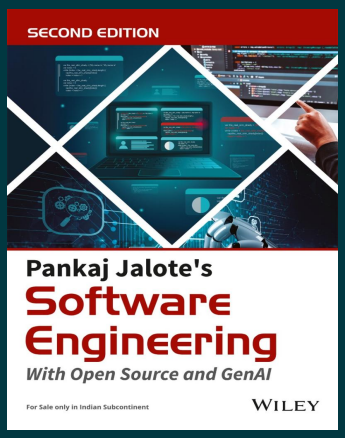
WILEY



Pankaj Jalote's Software Engineering: With Open Source and GenAI, 2ed

By Pankaj Jalote

Paperback 9789363869189 ₹750.00

Pages: 224

Description

In the rapidly evolving software development landscape, the integration of open-source tools and Large Language Models (LLMs) is essential. Pankaj Jalote’s Software Engineering: With Open Source and GenAI equips students with the skills to excel in managing and contributing to software projects. This book is perfect for Software Engineering courses at both undergraduate and graduate levels.

The book introduces key concepts such as industry-standard software, open-source tools, and the fundamentals of LLMs. It covers important areas like project management, development process models, and team collaboration. As students progress, they will explore requirements analysis, architecture design, coding, testing, and application deployment, all with a focus on real-world applications.

This textbook provides practical methods, real-world examples, and a strong emphasis on leveraging open-source software and the application of LLMs in modern software development.

This textbook is perfect for faculties aiming to provide students with practical skills and knowledge to manage modern software projects using cutting-edge technologies.

About the Author

Pankaj Jalote

Pankaj Jalote is a Distinguished Professor at the Indraprastha Institute of Information Technology Delhi (IIIT-Delhi an institution where he also played a pivotal role as the Founding Director. His academic journey includes prestigious positions such as Chair Professor at the Indian Institute of Technology Delhi, Head of the Department of Computer Science and Engineering at IIT Kanpur, and an early career role as an Assistant Professor at the University of Maryland, College Park. Beyond academia, he has made his mark in the industry as the Vice President of Infosys Technologies Ltd. for two years and as a Visiting Researcher at Microsoft Corporation in Redmond for a year. His educational background is as impressive as his career, holding a BTech from IIT Kanpur, an MS from Pennsylvania State University, and a PhD from the University of Illinois at Urbana-Champaign.

He has authored six books. Among these is the highly regarded "CMM in Practice," which has been translated into multiple languages, including Chinese, Japanese, and Korean, and the widely used textbook "An Integrated Approach to Software Engineering." His contributions to the field have earned him recognition as a Fellow of the IEEE and the Indian National Academy of Engineering (INAE), and he has been honored with the Distinguished Alumnus Award from IIT Kanpur.

Table of Contents

About the Author

Preface
1. Industry-Strength Software
1.1 Industry-Strength Software
1.1.1 Demo and Industry-Strength Software
1.1.2 Types of Software
1.2 Key Drivers: Productivity and Quality
1.2.1 Productivity
1.2.2 Quality
1.2.3 Improving Productivity and Quality
1.3 Open-Source Software (OSS)
1.3.1 Evolution of Open-Source Software (OSS) and Current State
1.3.2 Libraries and Frameworks
1.3.3 Open-Source Software Licensing
1.4 LLMs and Prompt Engineering
1.4.1 Single Prompt Approaches
1.4.2 Multi-Prompt Approaches
Summary
Self-Assessment Exercises
2 Planning a Software Project
2.1 Software Development Process
2.1.1 Waterfall Model
2.1.2 Prototyping
2.1.3 Iterative Development
2.1.4 Agile Process
2.1.5 Open-Source Software (OSS) Process
2.2 Project Management
2.2.1 Traditional Planning and Management
2.2.2 Open-Source Development
2.3 Team-Working
2.3.1 Effective Teams
2.3.2 Pillars of Teamwork
2.3.3 Teamwork in Practice
Summary
Self-Assessment Exercises
3 Software Requirements Analysis and Specification
3.1 Requirement Process
3.1.1 Problem Analysis
3.1.2 Specification

3.1.3 Validation
3.2 Requirements Specification
3.2.1 Desirable Characteristics of a Software Requirements Specification (SRS)
3.2.2 Functional and Non-Functional Requirements
3.2.3 Structure of a Requirements Document
3.2.4 Minimum Viable Product (MVP)
3.2.5 Requirements in Open-Source Process
3.3 Functional Specification With Use Cases
3.3.1 Basics
3.3.2 Examples
3.3.3 Developing Use Cases
3.3.4 User Stories
3.4 An Example SRS
3.5 Using LLMs for Requirements
3.5.1 Generating the Software Requirements Specification (SRS)
3.5.2 Software Requirements Specification (SRS) Validation
Summary
Self-Assessment Exercises
4 Software Architecture
4.1 Architecture Views
4.2 Component and Connector (C&C) Architecture
4.2.1 Components
4.2.2 Connectors
4.2.3 An Example
4.3 Architectural Styles
4.3.1 Pipe and Filter
4.3.2 Shared-Data
4.3.3 Client-Server
4.3.4 3-Tier
4.3.5 Model-View-Template (MVT) and Model-View-Controller (MVC)
4.3.6 Microservices
4.3.7 Some Other Styles
4.4 Architecture Design Decisions
4.4.1 Architectural Style
4.4.2 Technology Stack and Open-Source Choices
4.4.3 Other Considerations (Security, Scalability, etc.)
4.4.4 Using LLMs for Architectural Decisions

4.4.5 Documenting and Evaluating Architecture
4.5 Example: Architectural Design
Summary
Self-Assessment Exercises
5 Design
5.1 Design Concepts
5.1.1 Coupling
5.1.2 Cohesion
5.1.3 The Open-Closed Principle
5.2 Design Methodologies
5.2.1 Function-Oriented Structured Design
5.2.2 UML and Object-Oriented (OO) Design
5.3 Designing an Application
5.3.1 Front-End Design
5.3.2 API-Based Back-End Design
5.3.3 Model Layer Design
5.4 Example: Overview of the Student-Event Portal Application Design
5.5 Using LLMs for Design
Summary
Self-Assessment Exercises
6 Coding
6.1 Programming Principles and Guidelines
6.1.1 Structured Programming
6.1.2 Information Hiding
6.1.3 Pre and Post Conditions
6.1.4 Coding Standards and Guidelines
6.1.5 Using Open-Source Libraries
6.2 Managing Code
6.2.1 Code Organisation
6.2.2 Source Code Control
6.3 Developing Code
6.3.1 An Incremental Coding Process
6.3.2 Test-Driven Development (TDD)
6.3.3 Pair Programming
6.4 Using LLMs for Coding
6.4.1 Using Copilot
6.4.2 Using ChatGPT
6.5 Verification

6.5.1 Unit Testing
6.5.2 Code Reviews
6.6 Metrics
6.6.1 Size Measures
6.6.2 Complexity Metrics
Summary
Self-Assessment Exercises
7 Testing
7.1 Testing Concepts
7.1.1 Failure, Fault, Error and Limits of Testing
7.1.2 Test Case, Test Suite, and Test Scripts
7.1.3 Psychology of Testing
7.1.4 Levels of Testing
7.1.5 Test Plan
7.2 Test Case Design
7.2.1 Black-Box Testing
7.2.2 White-Box Testing
7.3 Using LLMs For Generating Test Cases
7.3.1 Generating Test Scripts for Unit Testing
7.3.2 Generating Test Cases for System Testing
7.4 Test Case Execution
7.4.1 Using Open-Source Testing Tools and Frameworks
7.4.2 Defect Logging and Tracking
7.5 Metrics
7.5.1 Coverage Analysis
7.5.2 Defects Data
7.5.3 Reliability
7.5.4 Defect Removal Efficiency
Summary
Self-Assessment Exercises
8 Application Deployment
8.1 The Deployment Process
8.2 Release and Packaging
8.2.1 Packaging Installable Applications
8.2.2 Packaging Web Applications
8.3 Installation and Activation
8.3.1 Installable Applications
8.3.2 Installing Web Applications

8.4 Using LLMs for Deployment 178

Summary 179

Self-Assessment Exercises

Frequently Asked Questions

References

Index

To purchase this product, please visit:
https://www.wileyindia.com/pankaj-jalote-s-software-engineering-with-open-source-and-genai-2ed.html

Scan to buy