Continuous Integration and Delivery with Test-driven Development

Cultivating quality, speed, and collaboration through automated pipelines

Amit Bhanushali

Alekhya Achanta

Beena Bhanushali

www.bpbonline.com

First Edition 2024

ISBN: 978-93-55519-726

LIMITS OF LIABILITY AND DISCLAIMER OF WARRANTY

The information contained in this book is true to correct and the best of author's and publisher's knowledge. The author has made every effort to ensure the accuracy of these publications, but publisher cannot be held responsible for any loss or damage arising from any information in this book.

All trademarks referred to in the book are acknowledged as properties of their respective owners but BPB Publications cannot guarantee the accuracy of this information.

Dedicated to

Our loving parents and teachers who taught us the basics of life and science

About the Authors

- **Amit Bhanushali**, a seasoned Quality Assurance Manager with 22 years of expertise, has excelled in Software Quality Optimization, particularly in the BFSI and higher education sectors. His proficiency spans automation testing, performance testing, and navigating complex DevOps and CI/CD environments, integrating cutting-edge technologies like AI and ML. With a Master's degree in Business Data Analytics from West Virginia University, Amit seamlessly combines academic insights with practical acumen. His impactful collaborations with Fortune 500 companies showcase a transformative blend of theoretical knowledge and hands-on experience. As a leader at West Virginia University, he has successfully spearheaded projects, reducing costs and enhancing education quality. Beyond his managerial role, Amit has authored research papers and novels on Software Quality Optimization, Automation Testing, AI, and ML. Recognized with the Innovator of the Year award at the Globee Business Awards 2023, his journey epitomizes innovation, leadership, and enduring transformation, making him a deserving recipient of the International Achiever Award.
- **Alekhya Achanta** is a Senior DataOps Engineer with expertise in BI, visualization, and data-driven decision-making. She creates robust data pipelines, dashboards, and actionable insights that optimize business outcomes. She is proficient in Python, SQL, data visualization, and tools like Matillion, DBT, and Power BI. She has an MS in Data Science, published 10+ scholarly papers in leading international journals, and recognized as a Top Voice on LinkedIn in Data Science. She received numerous accolades for the companies she worked with across the years for her intellectual curiosity and passion for problem-solving. With her strong technical expertise coupled with an ability to understand business needs, she delivers cutting-edge data solutions that create

real impact. Alekhya mentors ADPList and is a proud IEEE Senior Member.

- **Beena Bhanushali** is a highly skilled Salesforce CRM Administrator with a focus on optimizing Salesforce implementations for business growth. Specializing in CI/CD methodologies within the Salesforce ecosystem, Beena excels in streamlining processes and enhancing user experiences. With a keen eye for detail and problem-solving abilities, she collaborates effectively with cross-functional teams to deliver scalable solutions that meet clients' unique needs. Known for her commitment to staying abreast of industry trends, Beena is recognized as a trusted expert in Salesforce, making her an invaluable asset to any project or team.

About the Reviewers

- **Shantanu Neema** is an accomplished data scientist, recognized for delivering impactful insights in diverse industries through data-driven methodologies. With a proficiency in managing and analyzing datasets to define precise business use cases, he excels in crafting solutions for intricate challenges spanning real estate, energy, transportation, environmental compliance, and manufacturing. Shantanu's extensive experience encompasses the entire data science process, culminating in model deployment using cloud infrastructure. His expertise extends to a robust foundation in CI/CD, ML pipelines, and testing methodologies, ensuring the efficiency and resilience of his solutions. Beyond his technical role, Shantanu actively engages as a researcher and serves as a technical reviewer for books centered around CI/CD, data science, and Python. This commitment underscores his dedication to advancing best practices and fostering innovation in these dynamic fields. Shantanu Neema invites readers to explore his insights and contributions, encapsulated within the pages of publications that reflect his ongoing pursuit of excellence in data science and technology.

- **Dr. Nalini Jagtap** is a distinguished academician with over 8 years of teaching experience, serving as an Associate Professor at Dr D Y Patil Institute of Engineering Management and Research. She holds a Doctorate in Computer Engineering from Savitribai Phule Pune University, where she was ranked first in her Master's program. Her dedication to academic excellence is truly commendable.

  Dr. Nalini Jagtap's impressive background in academia is underscored by remarkable achievements in research and teaching. With expertise spanning Computer Vision and Pattern Recognition, as well as Artificial Intelligence and Machine Learning, Dr. Jagtap has established herself as a leading figure in these fields. Demonstrating exceptional teaching prowess throughout her 8

years of experience, Dr. Jagtap has imparted knowledge across various subjects.

In addition to her academic endeavors, Dr. Jagtap has made substantial contributions to the IT industry, showcasing her versatility and adaptability over 7.5 years in various roles. She combines academic excellence with extensive practical experience. Beyond her academic and professional accomplishments, Dr. Jagtap has significantly contributed to the advancement of her field through influential research papers, a testament to her deep-rooted background in research.

Acknowledgement

We are grateful to BPB Publications for their guidance and expertise in bringing this book to fruition. Revising this book was a long journey, with valuable participation and collaboration of reviewers, technical experts, and editors.

We would also like to acknowledge the valuable contributions of our mentors and colleagues during many years working in the tech industry, who have taught us so much and provided valuable feedback on our work.

Special gratitude is extended to Dr. Nalini Jagtap and Shantanu Neema for their invaluable contributions as technical reviewers. Dr. Nalini Jagtap, with over 8 years of teaching experience and a Doctorate in Computer Engineering, provided insightful feedback reflecting her dedication to academic excellence, particularly in computer vision and AI. Meanwhile, Shantanu Neema, an accomplished Data Scientist, demonstrated his expertise in managing and analyzing datasets across diverse industries, including real estate, energy, transportation, environmental compliance, and manufacturing. His proficiency in model deployment on cloud infrastructure and implementation of CI/CD, ML pipelines, and testing methodologies greatly enhanced the quality of the work. Their combined efforts significantly enriched the research and academic endeavors.

Finally, we would like to thank all the readers who have taken an interest in our book and for their support in making it a reality. Your encouragement has been invaluable.

Preface

The landscape of software development has transformed radically, with customer expectations for faster delivery of high-quality digital products intensifying exponentially. Much as strict protocols govern safety-critical systems, today's complex web and mobile ecosystems demand stringent quality practices underpinned by comprehensive testing. Yet many resources focus excessively on theory without addressing practical application.

This hands-on guide bridges that gap, offering practitioners an invaluable inside understanding of how leading organizations optimize software and data CI/CD pipelines to accelerate release cycles without compromising stability or user experience. Balancing cutting-edge technical foundations with indispensable cultural transformation, the book equips enterprises to actualize the DevOps mandate of fail-fast innovation, seamless collaboration, and ruthless automation.

With continuous practices now an indispensable pillar of IT strategy, these pages detail battle-tested frameworks for quality engineering tailored to modern release trains. Through expert coverage of must-have toolchains as well as processes that safeguard both velocity and verification, readers will grasp not only CI/CD's immense potential but also its practical implementation and governance.

Complimenting conceptual mastery with actionable playbooks, this book illuminates the synergy between lean culture, behavioral best practices, and optimized pipelines. Readers will gain unprecedented clarity into the real-world changes necessary to retool release processes, test automation, and team dynamics that many Continuous Delivery initiatives overlook to their detriment.

Whether a novice seeking fundamental fluency or a leader charging towards DevOps excellence, this guide delivers the definitive reference for unlocking CI/CD's total value. The future of software lies in empathy, quality, and flow; it is our privilege to light the path forward.

Chapter 1: Adopting a Test-driven Development Mindset – Testing is an integral part of the **software development lifecycle (SDLC)**. As IT professionals, adopting a test-driven mindset enables us to deliver higher quality software through rapid feedback loops. In this chapter, we introduce the readers to **test-driven development (TDD)** methodology and contrast it with traditional testing approaches. Furthermore, we delve into the significance of data in modern software development and introduce the concept of DataOps, which emphasizes the importance of data operations in the agile development process.

Chapter 2: Understanding CI/CD Concepts – This chapter discusses the fundamental concepts of **continuous integration and continuous delivery (CI/CD)**, as well as the emerging paradigm of data CI/CD. Exploring the principles and benefits of CI/CD and data CI/CD, readers will gain a comprehensive understanding of how these practices streamline software and data workflows. They will learn how CI/CD practices enhance software quality while data CI/CD focuses on ensuring data integrity, quality, and reliability. By the end of this chapter, readers will be prepared to embrace both CI/CD and data CI/CD as integral components of modern software and data development, fostering efficiency, collaboration, and quality.

Chapter 3: Building the CI/CD Pipeline – This chapter discusses the intricacies of CI, CD, and data CI/CD, emphasizing their pivotal role in contemporary software development. We will guide you through the meticulous process of crafting a resilient CI/CD pipeline that integrates code and ensures seamless delivery of data. From the initial stages of code writing and testing to the final product delivery, we illuminate the critical elements that bolster code quality, uniformity, and swift deployment. By harnessing the power of this comprehensive system, development teams can significantly reduce manual interventions, leading to minimized errors and expedited results. With the added dimension of data CI/CD, we ensure that applications are always fueled by the most current and accurate data, enhancing overall productivity.

Chapter 4: Ensuring Effective CD – In this chapter, we cover the critical aspects of CD, focusing on both software and data. We explore

topics related to real-time monitoring, observability practices, robust security measures, and strategic release management. By mastering the content of this chapter, readers will be well-equipped to navigate the complexities of modern CD, ensuring not only the seamless deployment of software but also safeguarding its integrity, optimizing performance, and ensuring compliance with industry standards.

Chapter 5: Optimizing CI/CD Practices – From addressing the unique demands of expansive projects and diverse environments to fostering an unyielding commitment to continuous improvement, this chapter equips readers to elevate their CI/CD endeavors. By absorbing the insights and strategies offered within these pages, readers are primed to optimize their CI/CD practices for both software and data, ensuring smoother delivery and consistent refinement of processes and outcomes.

Chapter 6: Specialized CI/CD Applications – From delving into mobile and IoT contexts to leveraging an arsenal of tools and embracing best practices, this chapter equips readers to navigate the terrain of specialized CI/CD domains. Armed with the insights garnered from this chapter, readers will be poised to implement CI/CD solutions tailored to distinct contexts, fostering efficiency, innovation, and excellence.

Chapter 7: Model Operations: DevOps Pipeline Case Studies – This chapter presents a collection of case studies that shed light on real-world applications of CI/CD practices. By examining these cases, readers will glean insights into how various organizations and projects have harnessed CI/CD to achieve remarkable outcomes, fostering innovation, reliability, and accelerated delivery.

Chapter 8: Data CI/CD: Emerging Trends and Roles – In closing, while the technical disciplines of CI/CD are essential, it is also vital that organizations nurture a collaborative culture focused on software quality, speed, and responsiveness to customer needs. Technical professionals should advocate for and model these values. By combining automated pipelines with cultural transformation, IT organizations can unlock the full benefits of CI/CD.

Code Bundle and Coloured Images

Please follow the link to download the
Code Bundle and the ***Coloured Images*** of the book:

https://rebrand.ly/hm0s5m1

The code bundle for the book is also hosted on GitHub at

https://github.com/bpbpublications/Continuous-Integration-and-Delivery-with-Test-driven-Development
In case there's an update to the code, it will be updated on the existing GitHub repository.

We have code bundles from our rich catalogue of books and videos available at **https://github.com/bpbpublications**. Check them out!

Errata

We take immense pride in our work at BPB Publications and follow best practices to ensure the accuracy of our content to provide with an indulging reading experience to our subscribers. Our readers are our mirrors, and we use their inputs to reflect and improve upon human errors, if any, that may have occurred during the publishing processes involved. To let us maintain the quality and help us reach out to any readers who might be having difficulties due to any unforeseen errors, please write to us at :

errata@bpbonline.com

Your support, suggestions and feedbacks are highly appreciated by the BPB Publications' Family.

Piracy

If you come across any illegal copies of our works in any form on the internet, we would be grateful if you would provide us with the location address or website name. Please contact us at **business@bpbonline.com** with a link to the material.

If you are interested in becoming an author

If there is a topic that you have expertise in, and you are interested in either writing or contributing to a book, please visit **www.bpbonline.com**. We have worked with thousands of developers and tech professionals, just like you, to help them share their insights with the global tech community. You can make a general application, apply for a specific hot topic that we are recruiting an author for, or submit your own idea.

Reviews

Please leave a review. Once you have read and used this book, why not leave a review on the site that you purchased it from? Potential readers can then see and use your unbiased opinion to make purchase decisions. We at BPB can understand what you think about our products, and our authors can see your feedback on their book. Thank you!

For more information about BPB, please visit **www.bpbonline.com**.

Table of Contents

Chapter 1
Adopting a Test-driven Development Mindset

Introduction

The purpose of this chapter is to define different approaches to software testing and to outline the pros and cons of each approach. Software development and testing are integral and closely related aspects of **software development lifecycle** (**SDLC**).

As applications become more data-centric, the interplay between software development, testing, and data management becomes even more critical. This evolving landscape introduces new challenges and considerations, hinting at the emergence of specialized methodologies like DataOps.

The software development method for a particular project will influence the choice of testing methodology. The two primary development approaches are briefly outlined here to facilitate deciding which direction a particular project will take.

Structure

The chapter covers the following topics:

- Traditional methodology
- Agile methodology
- Development
- Weighing the pros and cons
- Evolving role of data in software development
- Introduction to DataOps

Objectives

Testing is an indispensable component of the software development process. As IT professionals, adopting a test-driven mindset enables us to deliver higher-quality software through rapid feedback loops. In this chapter, we introduce the readers to **test-driven development** (**TDD**) methodology and contrast it with traditional testing approaches. Furthermore, we delve into the significance of data in modern software development and introduce the concept of DataOps, which emphasizes the importance of data operations in the agile development process.

Traditional methodology

The most common software development techniques are the **waterfall model**, **spiral model**, and **V model**. Though all of them have different concepts, they all have one thing in common, that is, the test is executed only after coding. The waterfall model is a sequential, linear approach to software design. The waterfall model is a strictly linear progression that moves through various stages, including conception, initiation, analysis, design, construction, testing, implementation, and maintenance. It is imperative to follow this process precisely to achieve optimal results. Any deviation from this model may lead to undesirable outcomes.

The waterfall methodology places significant emphasis on project planning, as it is crucial to have a precise plan and vision before commencing development. This approach's detailed planning allows the software to be launched quickly while providing greater accuracy in estimating budgets and timelines.

Agile methodology

The agile approach to software design is highly flexible, with adaptive planning and evolutionary development at its core. Agile is a freeform design model that involves developers working on small modules at a time. Throughout the development process, customer feedback and software testing happen simultaneously. This approach offers numerous benefits, particularly in project environments where development needs to be responsive and effective in the face of changing requirements.

This methodology promotes interaction and communication in software development, prioritizing collaboration over design to enable effective engagement between designers and stakeholders. It is especially beneficial in environments that emphasize teamwork. The development process involves multiple developers working on separate modules that are later integrated to produce a complete software product at the end of each iteration.

Development

In the dynamic world of software engineering, the term **development** encapsulates the processes, methodologies, and practices employed to bring software solutions to life. Development is a multifaceted journey from ideation to deployment that transforms requirements into functional software. Over the years, various approaches to development have emerged, each with its unique perspective on how software should be built and tested. Two prominent methodologies are **traditional development** and **TDD**. While the former emphasizes a sequential approach with testing following development, the latter intertwines testing with development, advocating for tests to be written even before the code itself.

Traditional development

A common practice in software development is to have an independent group of testers perform testing after the functionality is developed but before it is shipped to the customer. Refer to the following *Figure 1.1*:

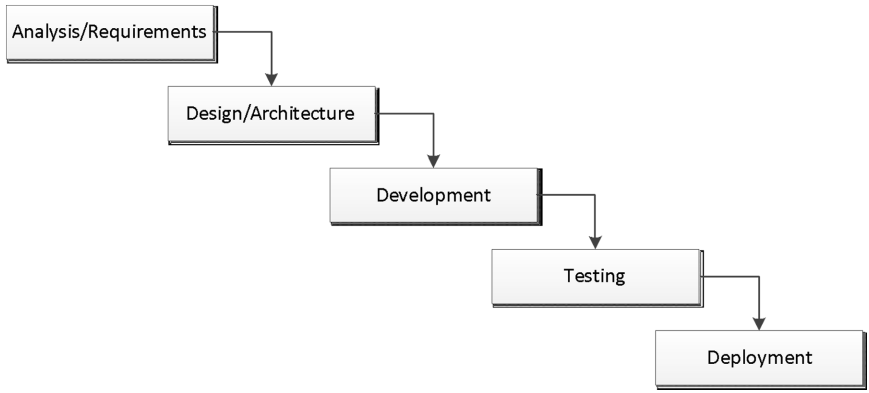

Figure 1.1: *Traditional development workflow*

Test-driven development

The initiation of software testing should commence at project commencement and endure until the ultimate completion phase. Refer to the following *Figure 1.2*:

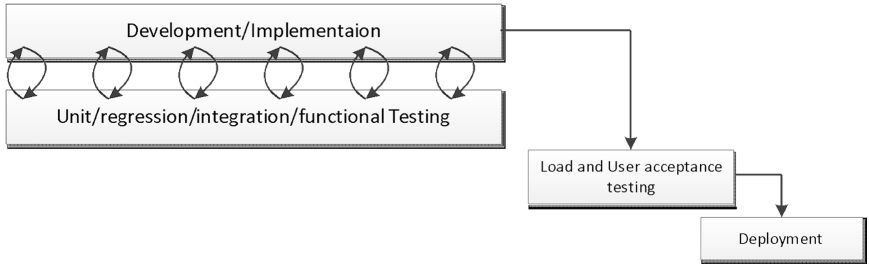

Figure 1.2: *TDD workflow*

Some software development methodologies, like Agile and Extreme programming, use a test-driven approach. Software engineers often write unit tests first. They frequently work in pairs and use the extreme programming method. At the start, engineers intentionally designed these tests to fail at first. However, as they write the code, it begins to pass. Over time, the code successfully meets more and more conditions of the test suites. The test suites are regularly updated with new failure conditions and corner cases and integrated with regression tests. Unit tests are maintained alongside the software code and