CI/CD *Anti-Patterns*

Zhimin Zhan

CI/CD Anti-Patterns

Lessons from Real-World CI/CD Failures

Zhimin Zhan

This book is available at https://leanpub.com/ci-cd-anti-patterns

This version was published on 2026-01-19

Contents

Preface . . . . **i**

Purpose of the book ii

What is unique about this book? ii

Who the book is for? iii

Send me feedback iii

1: Introduction **1**

1.1: Continuous Integration (CI) 1

1.2: Continuous Delivery and CI/CD 7

1.3: Who does "CI/CD"? 8

1.4: 80+% of CI/CD effort is about Test Automation 9

1.5: Continuous Testing 9

2: No Real Delivery **13**

2.1: Anti-Pattern: Unclear on delivery destination 13

2.2: Anti-Pattern: Fear to deploy to a test-ready environment 13

2.3: Anti-Pattern: Auto production deployment 13

2.4: Anti-Pattern: CI/CD is disconnected from production releases 13

2.5: Wrap-up 13

3: Non-Testing CI Steps **15**

3.1: Anti-Pattern: Not fully-automated CI steps 15

3.2: Anti-Pattern: Manual database schema migration 15

3.3: Anti-Pattern: CI Engineers lack automation skills 15

3.4: Anti-Pattern: Slow compile/build/package 15

3.5: Anti-Pattern: Fixated with code coverage . . . 15
3.6: Anti-Pattern: No auto-deployment to internal server environments . . 16
3.7: Anti-Pattern: Lack of notification to the team about CI/CD build results 16

4: Test Automation is the Core of CI/CD . . . 17
4.1: Anti-Pattern: No Automated Testing in CI . . . 17
4.2: Anti-Pattern: Automated Test Execution Not Enabled in CI . . . 17
4.3: Anti-Pattern: Remove failed automated tests to pass the run . . . 17
4.4: Anti-Pattern: Long-Running Unit Tests in CI . . . 17
4.5: Anti-Pattern: Frequent Unit Test Failures in CI . . . 17
4.6: Anti-Pattern: Long-Running Integration Tests in CI . . . 18
4.7: Anti-Pattern: Automated Tests Not Run Often in CI . . . 18
4.8: Anti-Pattern: No Regression Testing in CI/CD . . . 18
4.9: Wrap-up . . . 18

5: E2E Test Automation . . . 19
5.1: Anti-Pattern: No Automated Acceptance Testing in CI/CD . . . 19
5.2: Anti-Pattern: Sequential Executing E2E Tests in CI/CD . . . 19
5.3: Anti-Pattern: Avoid Running the Entire Suite—Split It into Smaller Suites 19
5.4: Anti-Pattern: Prioritizing Tests in Execution . . . 19
5.5: Anti-Pattern: Fake Automated Acceptance Testing . . . 20
5.6: Anti-Pattern: E2E Test Scripts are Not in an Independent Git Repository 20
5.7: Anti-Pattern: Headless Browser Testing . . . 20
5.8: Anti-Pattern: Long execution time > 1 hour . . . 21
5.9: Anti-Pattern: Parallel Execution on one machine . . . 21
5.10: Wrap-up . . . 22

6: CT Features Missing in Most CI/CD Servers . . . 23
6.1: View Test Scripts in the CT Server Web Interface . . . 23
6.2: Capture Application Screenshots on E2E Test Failure . . . 23
6.3: Quick navigation to the failed test line . . . 23
6.4: Auto-Retry Failed E2E Test on Another Build Agent . . . 23
6.5: Manually Rerun Tests . . . 23
6.6: Dynamic Test Execution Ordering . . . 24

6.7: Distribution rules 24
6.8: Delay completion 24
6.9: Anti-Pattern: Being constrained by a vendor's CI/CD server limitations and do nothing 24
6.10: Wrap-up 24

7: Mindset **25**
7.1: Anti-Pattern: "Test Automation in CI Slows Down Development" 25
7.2: Anti-Pattern: "Prefer expensive commercial CI/CD solutions over free open-source ones" 25
7.3: Anti-Pattern: "API Testing as E2E Test Automation in CI/CD is Enough" 25
7.4: Anti-Pattern: "E2E Test Automation Is Inherently Fragile, Keep Them To a Minimum" 25
7.5: Anti-Pattern: "E2E Test Automation Is Too Expensive, So we don't include in CI/CD" 26
7.6: Anti-Pattern: Assuming Automated Tests Will Always Pass After Creation 26
7.7: Anti-Pattern: "E2E Test Automation Scripts Must Use the Same Language as the Application Code" 26
7.8: Anti-Pattern: "Set up a CI/CD Solution Takes a Long Time and Costly" 26
7.9: Anti-Pattern: Drawn to flashy, smoke-and-mirror features like Test Replay 27
7.10: Wrap-up 27

8: General Practices **29**
8.1: Anti-Pattern: Not the exact the same release package for different target environments. 29
8.2: Anti-Pattern: Underinvesting in CI/CD Hardware 29
8.3: Anti-Pattern: Automating the Tear Down and Setup of the CI Servers Daily 29
8.4: Anti-Pattern: No Admin Access for Developers and Testers in CI/CD . 29
8.5: Anti-Pattern: Only limited to execute one specific type of automated tests in one language 29

8.6: Anti-Pattern: A Single CI/CD Server Serving the Entire Company/-Division . . . 30
8.7: Anti-Pattern: No Daily Builds . . . 30
8.8: Anti-Pattern: No incremental version or build number during internal testing . . . 30
8.9: Anti-Pattern: Commit-Driven CI/CD Triggers . . . 30
8.10: Anti-Pattern: Sending Real Emails Out During Test execution in CI/CD 30

9: Non-Functional Testing . . . 32
9.1: Anti-Pattern: No Continuous Performance Testing . . . 32
9.2: Anti-Pattern: No Continuous Load Testing . . . 32
9.3: Wrap-up . . . 32

10: CI/CD Hosting . . . 34
10.1: Anti-Pattern: Neglecting the Needs of E2E Test Automation . . . 34
10.2: Anti-Pattern: Falling for flashy but useless features in cloud testing platforms but miss the big picture . . . 34
10.3: Anti-Pattern: Completely unaware of or blindly exlcuding on-premise Setup . . . 34
10.4: Wrap-up . . . 37

11: Team Involvement . . . 38
11.1: Execute Individual Tests via the CT Server Web Interface . . . 38
11.2: Data Preparation for User Acceptance Testing . . . 38
11.3: Self-help Utilities that whole team can use on the CT Server . . . 38
11.4: Generate a Traceability Matrix . . . 38
11.5: Anti-Pattern: Non-Tech Team Members Not Using CI/CD Server . . . 39
11.6: Anti-Pattern: Only One Run Configuration . . . 39
11.7: Case Study: A bug-fixing cycle in a team with real CI/CD . . . 39
11.8: Real CI/CD Boots Customers' Trust and Confidence in the team . . . 39

12: Leadership . . . 41
12.1: Anti-Pattern: Failing to Recognize the Great Benefits of Real CI/CD . 41
12.2: Anti-Pattern: Lack Team-Wide Training on CI/CD Process . . . 41

12.3: Anti-Pattern: Developers delay fixing newly discovered defects by Continuous Testing 41
12.4: Anti-Pattern: Stuck with One CI Vendor or One Person 41
12.5: Anti-Pattern: Assuming Existing Staff Can Deliver Real CI/CD and E2E Automation with a New Product 42
12.6: Anti-Pattern: Fixated with All-in-one CI/CD. Two Stage CI □ CD Often Better 42

Afterword **43**
My CI/CD Formula 43
How many E2E Tests in CI/CD are enough? 43
My Continuous Testing Lab Setup 43

Appendix I: Anti-Patterns List **44**

Appendix II: CI Steps **45**

Resources **46**
Books 46
Tools 46
Blog 47

References **48**

Preface

Continuous Integration and Continuous Delivery (CI/CD) are among the most talked-about practices in modern software engineering, and also among the most misunderstood. From the very beginning, the original idea of Continuous Integration (introduced by Martin Fowler in 2000 [Fowler00]) implicitly assumed frequent, production-ready deployment.

A decade later, the publication of Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation [HumbleFarley10], the influential book by Jez Humble and David Farley in 2010, which won the prestigious Jolt Excellence Award the following year, brought the term Continuous Delivery into the mainstream. Somewhere along the way, "CI" quietly became "CI/CD".

Yet, another 15 years later, genuine delivery from CI/CD pipelines in software projects remains extremely rare. Nearly every software company claims to be "doing CI/CD," yet many still struggle with slow releases, fragile pipelines, late discovery of defects, and a general lack of confidence in the software packages produced by the CI/CD pipeline, let alone deploying them to production.

In reality, many software teams end up with so-called "CI/CD pipelines" that are technically sophisticated but operationally ineffective: look impressive in diagrams yet actively work against fast, safe delivery. These are rarely failures of intent; they result from a combination of incompetence, resistance to learning, and systemic anti-patterns that accumulate over time.

An anti-pattern is dangerous because it often appears sensible at first. It can arise as a shortcut under time pressure, a compromise to "keep things moving," a mistaken decision by a senior engineer, the wrong choice of test automation framework or CI/CD tool, or a small tweak that unintentionally creates bigger problems. In CI/CD, such anti-patterns are common and many, as you will see in this book. Unfortunately, once one CI/CD anti-pattern takes hold, teams often struggle to remove evenfully recognizes. Most simply live with it, painfully.

Purpose of the book

The purpose of this book is straightforward: to help software teams recognize, understand, and avoid the anti-patterns that undermine CI/CD. Unlike guides that focus on tools, frameworks, or the latest buzzwords, this book addresses the underlying practices and decisions that make CI/CD pipelines ineffective, fragile, or merely superficial.

What is unique about this book?

The primary reason most CI/CD pipelines fail is the absence of comprehensive test automation—especially end-to-end (E2E) UI automation used as regression testing. Improving a team's productivity only has real value when releases are properly qualified through solid testing; otherwise, the outcome is often counterproductive. Notably, this topic is largely avoided in most CI/CD books.

E2E (UI) test automation itself is a major challenge and is riddled with anti-patterns (as discussed in my *End-to-end Test Automation Anti-Patterns* book). One reality many software professionals fail to recognize is that E2E test automation and CI/CD form a Yin–Yang relationship:

- E2E test automation (as regression testing) is what makes CI/CD credible. It provides fast, trustworthy feedback and ensures that only production-ready releases leave the pipeline.
- An effective CI/CD setup, in turn, alleviates the core challenges of E2E automation, such as reliability issues and long execution times.

I learned this through experience. In a government project I led in 2008, with strong support from the project manager, we adopted E2E test automation across the entire team. However, as the test suite grew to around 150 tests, it became increasingly difficult–and eventually nearly impossible–to achieve a clean run on CruiseControl, which at the time was the first and one of only two CI servers available (*the other was Hudson*).

Unable to find an existing solution, I designed and implemented a set of UI test execution capabilities (later called "CT features") as plugins for the open-source CruiseControl server. The impact was transformative: the team could not only develop more tests but, more importantly, sustainably maintain a large E2E regression suite and execute it daily on the CI server, enabling reliable production releases every few days.

Who the book is for?

This book is for software professionals involved in building, managing, or maintaining CI/CD pipelines. Whether you are a DevOps engineer, QA lead, or technical manager, it offers practical insights to help you identify and avoid the most common and costly CI/CD anti-patterns.

For software startup owners, solopreneurs, or side hustlers ready to embrace the "**Ship early, ship often**" product strategy, I also share my CI/CD tech stacks and practices (*which are universially appliable* that have proven successful in my app development (and for my clients) for more than a decade.

Send me feedback

I would like to hear from you. Comments, suggestions, errors in the book are all welcome. You can submit your feedback via the book website.

Zhimin Zhan

Brisbane, Australia

1: Introduction

Continuous Integration (CI), to my knowledge, started with Martin Fowler's original CI article (2000–09–10), which started with: "Software development is full of best practices which are often **talked** about but seem to be **rarely done**. One of the most **basic**, and **valuable**, of these is a fully **automated** build and **test** process that allows a team to build and test their software **many times a day**."

> "Frequent releases benefit the organization and, in practice, make work easier. When you can release at any time, life is simply better." – The Art of Agile Development (Shore & Warden, 2008, p. 210) [ShoreWarden08]

1.1: Continuous Integration (CI)

Continuous Integration (CI) originated with Extreme Programming (XP) as one of its core practice.

> "We are using the term Continuous Integration, a term used as one of the practices of XP (Extreme Programming)." - Martin Fowler's original CI article[1].

Younger generations of software professionals may be unfamiliar with XP, which started the agile movement with Kent Beck's classic book "Extreme Programming Explained: Embrace Change" (published in 1999). Many later "Agile" methodologies, such as Scrum, Kanban, and SAFe, do not match XP in clarity or completeness. have been making this point for decades, often facing disagreement. In 2019, I was glad

[1] https://martinfowler.com/articles/originalContinuousIntegration.html

to find my perspective affirmed in "Clean Agile: Back to Basics" (2019) by Robert C. Martin, co-author of Agile Manifesto.

> "of all he Agile processes, XP is the best defined, the most complete, and least muddled" - Robert C. Martin in "Clean Agile: Back to basic" [Martin19], page 31.

Recently, I have noticed a resurgence of XP, as many so-called Agile initiatives have failed in numerous companies. But this raises a question: why wasn't XP more widely adopted? I can think of the following reasons:

1. **Naming**

   IT executives and managers tend to dislike the terms "Extreme" and "Programming".

2. **Mandate specific and measurable practices**

   Examples include "Pair Programming", "Test-Driven Development", "On-site customer", "Continuous Integration", "10-minute builds" and "Daily Production Deployments". It is hard to fake XP.

My point in explaining XP, the origin of CI, is to highlight a critical aspect of CI/CD that most companies miss: the goal of CI is daily production deployments.

> "**Daily Deployment** Put new software in production every night." - Kent Beck in "Extreme Programming Explained: Embrace Change, 2n edition" [Beck04], chapter 9 "Corollary Practices", page 68.

This might change the way some software professionals view their work, right? Some may be skeptical. I can assure you that "daily production deployments" not only is this feasible (*I've been doing it for over 17 years, leading client projects and my own ones*), but I honestly cannot imagine any other way to do software development.

> “We have been told many times that this is a very difficult discipline to put in place, but once it is working, no one would ever think of going back to the old way of doing things.“ - Implementing Lean Software Development: From Concept to Cash [Poppendieck07], by Mary Poppendieck and Tom Poppendieck

Facebook was doing daily production releases even before the term ‘CI/CD’ existed.

> “Facebook is released twice a day, and keeping up this pace is at the heart of our culture. With this release pace, automated testing with Selenium is crucial to making sure everything works before being released.” – DAMIEN SERENI, Engineering Director at Facebook, at Selenium 2013 conference[2].

often recommend the excellent presentation ‘Continuous Integration at Facebook’ by Katie Coons, a software engineer in Facebook’s Product Stability division, delivered at the F8 Conference in 2016. Note that, at least as of 2015, Facebook was just using ‘CI’ rather than ‘CI/CD’.

So, CI does include comprehensive end-to-end testing (automated, of course) with a goal for production deployment.

1.1.1: CI Steps/Tasks

A CI process consists of a series of steps (or tasks). Below is my illustration of these steps from 2009, which I believe still holds true today.

[2] https://2014.seleniumconf.in/2013/speakers/index.html

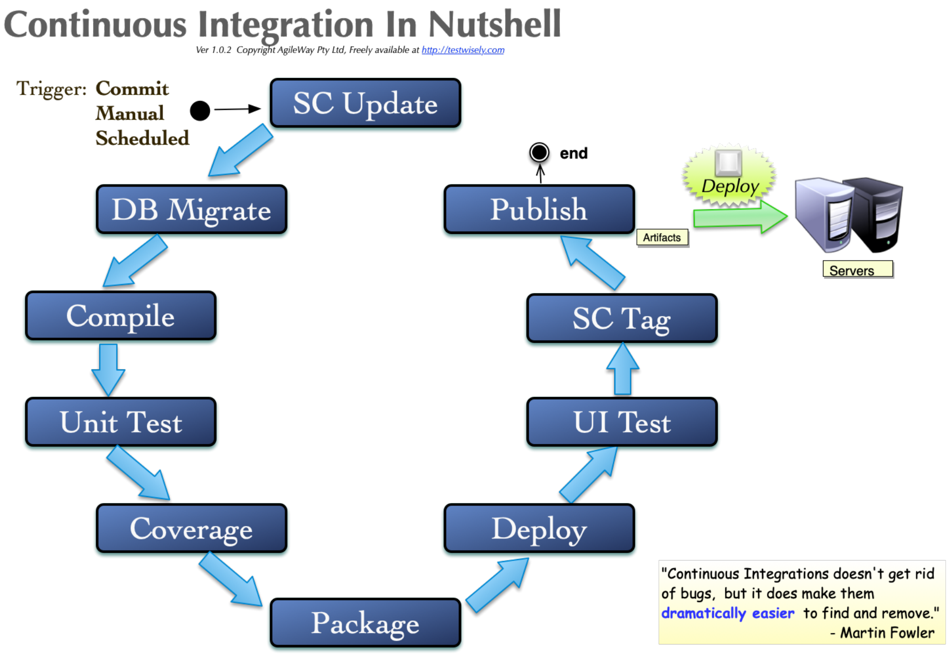

Continuous Integration In Nutshell
Ver 1.0.2 Copyright AgileWay Pty Ltd, Freely available at http://testwisely.com
Trigger: Commit
Manual
Scheduled
SC Update
DB Migrate
Compile
Unit Test
Coverage
Package
Deploy
UI Test
SC Tag
Publish
end
Artifacts
Deploy
Servers
"Continuous Integrations doesn't get rid of bugs, but it does make them dramatically easier to find and remove."
- Martin Fowler

Figure 1. CI in Nutshell

1. **Update source from a version control system** (Difficulty: *easy*)

   This is a built-in CI feature, and you don't usually need to do anything for this step, except installing the source control command line (or called client) tools.

2. **Database Migration** (Difficulty: *medium* or *hard*)

   During the life of software development, inevitably, your database schema needs changes along with the development, and in fact, quite frequently. For example, creating a new table or adding a new table column. This needs to be done systematically.

3. **Compile** (Difficulty: *easy*)

   This only applies to compiled languages such as Java or C#, You don't need this step if using a dynamic language such as Ruby.

4. **Unit Testing** (Difficulty: *hard*)

   Unit test, as the key concept of Test Driven Development, helps programmers to produce not only robust code, more importantly, but also better-designed code. If a piece of code is hard to write the unit test for it, its design is most likely not optimal.

   Programmers who claim 'refactoring code' without a suite of unit tests really are 'cow-boys change with hope for good luck'. No unit tests, no code refactoring.

5. **Code Coverage** (Difficulty: *medium*)

   Code coverage is measuring the percentage of code (in terms of methods/lines) that is covered by unit tests. This helps to find out untested and redundant code, to keep source code tidy and lean. It is also a good incentive for Test-Driven Development.

   In reality, achieving 100% code coverage is often not practical, 80% is already a good figure. As a matter of fact, you may find many projects without code coverage data or at a single-digit percentage.

   By including code coverage into the build process, we can help new/junior (not by age) programmers to develop the habit of writing unit tests.

6. **Package** (Difficulty: *medium*)

   A software release package typically contains compiled source code, configurations, web pages (with CSS and JavaScript), file templates, ..., etc. This step is mostly concerned with how to package files using build scripts into a specific format. For example, a war file is a zipped file format used for web applications developed in Java.

7. **Deploy** (Difficulty: *easy* or *medium*)

   Depending on the nature of your application, deployment can be quite complex (if heavyweight frameworks are used) or very simple (such as Ruby on Rails). A typical deployment process consists of the following steps:

   a) Stop the server if it is currently running

   b) Unpack the new release

   c) Update the database schema (database migration).
   d) Update configuration
   e) Start the server

   Deployment needs to be simple, reliable, and quick. With the popularity of cloud deployment, new deployment technology emerges such as Chef, Docker/Kubernetes containers. Unfortunately, many DevOps engineers (*by the way, I think it is a wrong title for a person who solely does deployment, as deployment is only counted for about 5% of DevOps work from my experiences*) don't use them well. More often than not, they make deployment over-complicated, and as a result, fragile and slow.

8. **Functional Testing** (Difficulty: *very hard*)

   Executing automated functional tests against the test server(s) with a new version of software deployed (by the last step), essentially, Continuous Testing.

   Some might not agree with the difficulty (**very hard**) I rated there.

   I would say we might have different perspectives. Most UI testing, if present, in CI is no more than smoke and mirrors. For me, CT is the core of software development. If all automated functional tests pass in CI, this build will be released to production. (For nearly all user stories and customer-found defects, I have automated tests for them).

   Let me illustrate it by an example. Let's say that we have 200 user-story-level functional tests written in Selenium WebDriver, on average, one test case has 30 test steps (each step represents a user operation, such as entering text, clicking a link and verifying certain text) and execution time of a single test is 30 seconds. In total, there are `30 x 200 = 6000` test steps and a full regression testing will take `6000` seconds. To get a green build (all tests pass), each of every 6000 test steps needs to pass within nearly 2 hours of test execution. A single failure results in a broken build.

   Now, do you agree this is a very, very hard task?

9. **Tag a build/release** (Difficulty: *easy*)

Tagging refers to labelling the repository at a certain point of time so that it can be easily retrieved in the future to achieve repeatable builds.

You don't have to label every build, probably only for the green builds (which have passed all tests). Be aware of the time difference between checking out from source control and actual tagging. A common approach is to introduce 'Code Freeze' or schedule full builds at night time.

10. **Publish** (Difficulty: *easy*)

    Once a build finishes, besides showing build status and other information on CI's web interface, there are numerous ways to publish the result, such as email, Slack, or even switching on a lava lamp.

    Apart from the success/failure indicator, build results might also highlight failed test cases and artifacts (files generated out of the build process). Most CI servers keep the build history, which can be generated into pretty charts for reporting purposes.

As you can see, there are three hard tasks (in the order of difficulty): "**Database Migration**", "**Unit Testing**" and "**Functional Testing**". Few will deny the needs of these steps, though most probably have never seen them done properly.

1.2: Continuous Delivery and CI/CD

The term Continuous Delivery stems from the book "Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation" by Jez Humble and David Farley [HumbleFarley10], published in 2010, won the 2011 Jolt Excellence Award. From its back cover: "This groundbreaking new book sets out the principles and technical practices that enable rapid, incremental delivery of high quality, valuable new functionality to **users**."

I highlight the `users`, this means CD's target is production deployment.

In an interview in October 2019[3], Lisa Crispin, the co-author of the Agile Testing book, said this: "They (Jez Humble and David Farley) asked me to be a technical

[3] https://www.infoq.com/articles/current-future-testing/

reviewer for their manuscript (the famous Continuous Delivery book) … I read it. It's a book about testing, you know, the whole book is really about testing. That's the heart of continuous delivery. Jez Humble is very supportive of my saying that".

So, CD really is a part of CI (see above) with stronger focus on delivery. In fact, there is no need for "CD" if CI is doing well. Its presence often signals that CI is widely poorly executed, with few real delivery benefits.

However, a total replacing "CI" with "CD" isn't ideal either, as it can diminish recognition for the CI work already done. This led to the creation of the hybrid term "CI/CD".

1.2.1: CI/CD Pipelines

It's common to see the suffix added to "CI/CD" to form "CI/CD Pipelines". To my knowledge, there was no term "CI pipelines" before.

To me, it feels like a marketing term. In a factory, a pipeline means fully automated production with a finished (*passing automated* QA *checks*) product ready to ship. In reality, the so-called "CI/CD pipelines" I have encountered rarely reach that standard.

1.3: Who does "CI/CD"?

CI/CD remains a relative new term in the software industry and is often not well implemented. Who performs this task, under which title, is often unclear. Based on my observation, CI/CD is typically configured and managed by a Software Architect or a DevOps Engineer.

"DevOps is a set of practices that combines software development (Dev) and IT operations (Ops). It aims to shorten the systems development life cycle and provide **continuous delivery** with high software **quality**.' - Wikipedia[4]

Clearly, DevOps is close tied with Continuous Delivery (CD) and testing (specially, automated acceptance testing). Name-wise, DevOps is not a good one as the

[4] https://en.wikipedia.org/wiki/DevOps

original meaning is largely lost. Today, nearly every software project claims to implement DevOps, and some even have roles titled “DevOps Engineer”. However, based on my observations over the past five years, most companies and projects are practicing what I would call “fake DevOps”. The reason, real “E2E Test Automation” and “CI/CD” are two fields are ‘rarely done well’ (*from original CI article*).

Readers may wonder, ‘Who should be responsible for CI/CD?’ My answer: the developers and testers within each team. I will discuss this topic in more detail in later chapters.

1.4: 80+% of CI/CD effort is about Test Automation

People are often surprised when I say, “More than 80% of CI/CD work is really about test automation” (You shouldn’t be, as earlier I already mentioned that the authors of the classic Continuous Delivery book clearly stated that testing is heart of CD). Once I explain why, most people get it.

The fact is that “**test**” and “**automated**” are included in the very short definition of CI. The only disagreement lies in the estimated proportion, whether it represents roughly 50% or 80%.

In the 10 steps (*see above*) of a typical CI pipeline, which ones do you think are essential and require effort? Most people would say “compile/build,” “testing,” and “deployment,” right? Well, among those three, the compile/build and deployment stages are usually static and fast. But build and deployment are mostly static and quick once configured. That’s why testing consumes the majority of CI effort.

So why are some people surprised by this? Because their CI pipelines are incomplete–often missing proper E2E test automation simply due to a lack of E2E test automation skills.

1.5: Continuous Testing

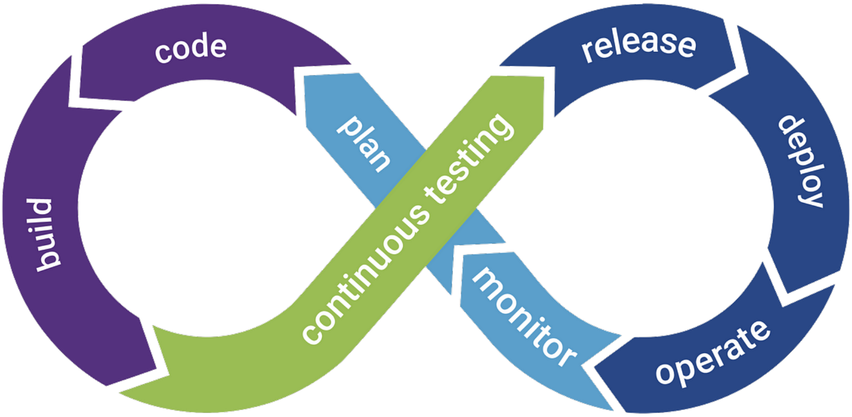
Because end-to-end testing is generally missing in most CI pipelines, the term Continuous Testing (CT) emerged to describe running automated E2E tests continuously throughout the process.

> “Continuous testing is “the process of executing automated tests as part of the software delivery pipeline to obtain immediate feedback on the business risks associated with a software release candidate.” - Wikipedia[5]

Continuous Testing is the key (and most important) process of DevOps.

> “By 2020, DevOps initiatives will cause 50% of enterprises to implement continuous testing using frameworks and open-source tools.” – Predicts 2017: Gartner Report[6]

You have probably seen various versions of the DevOps loop diagram.



[5] https://en.wikipedia.org/wiki/Continuous_testing
[6] https://www.gartner.com/en/documents/3525622

1.5.1: Continuous Testing (CT) vs CI/CD

If Continuous Integration (CI) were implemented properly, there would be no need for additional terms like Continuous Delivery (CD) or Continuous Testing (CT).

Unfortunately, CI has been so poorly executed in most software teams that the term has become almost meaningless. When "Continuous Delivery" emerged, people began pairing the two as "CI/CD," finding the phrase catchy, even though the substance was still lacking. In nearly every project I've seen over the past two decades, agile coaches and architects talked endlessly about CI/CD, yet rarely did any hands-on test automation. Unsurprisingly, their CI setups were embarrassing failures.

> I once visited a software company using a Bamboo CI server internally with multiple projects, yet the pipelines rarely ran and showed no signs of automated test execution. Still, they proudly claimed to be offering CI consultancy to one of Australia's top four banks.

If CI's main purpose is simply to build a releasable software package, that goal was already achieved long ago—before the CI concept even existed—using basic build scripts. For example, an Ant task generating a deployable WAR file back in the J2EE days. Triggering a build from a web interface and viewing results on a CI server is convenient, but is that really worth bragging about?

I use the terms CI, CD, CT interchangeably with its The true purpose of CI is to ensure quality production-ready releases by running automated tests against each release candidate. Testing, especially end-to-end, user-story-level testing, is the core of the process. The terminology simply places emphasis on different aspects:

- **CI** focuses more on build/packaging and unit testing.
- **CD** emphasizes producing high-quality, releasable outputs.

- **CT** highlights test automation, particularly automated end-to-end (UI) regression testing.

I named my CI/CD product BuildWise CT Server (internationally award-winning). It handles builds, packaging, and unit testing, but its real strength is executing large suites of automated end-to-end tests.

2: No Real Delivery

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

2.1: Anti-Pattern: Unclear on delivery destination

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

2.2: Anti-Pattern: Fear to deploy to a test-ready environment

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

2.3: Anti-Pattern: Auto production deployment

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

2.4: Anti-Pattern: CI/CD is disconnected from production releases

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

2.5: Wrap-up

3: Non-Testing CI Steps

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

3.1: Anti-Pattern: Not fully-automated CI steps

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

3.2: Anti-Pattern: Manual database schema migration

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

3.3: Anti-Pattern: CI Engineers lack automation skills

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

3.4: Anti-Pattern: Slow compile/build/package

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

3.5: Anti-Pattern: Fixated with code coverage

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

3.6: Anti-Pattern: No auto-deployment to internal server environments

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

3.7: Anti-Pattern: Lack of notification to the team about CI/CD build results

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

4: Test Automation is the Core of CI/CD

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

4.1: Anti-Pattern: No Automated Testing in CI

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

4.2: Anti-Pattern: Automated Test Execution Not Enabled in CI

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

4.3: Anti-Pattern: Remove failed automated tests to pass the run

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

4.4: Anti-Pattern: Long-Running Unit Tests in CI

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

4.5: Anti-Pattern: Frequent Unit Test Failures in CI

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

4.6: Anti-Pattern: Long-Running Integration Tests in CI

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

4.7: Anti-Pattern: Automated Tests Not Run Often in CI

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

4.8: Anti-Pattern: No Regression Testing in CI/CD

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

4.8.1: Regression Testing must be automated, however, …

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

4.9: Wrap-up

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

5: E2E Test Automation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

5.1: Anti-Pattern: No Automated Acceptance Testing in CI/CD

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

5.2: Anti-Pattern: Sequential Executing E2E Tests in CI/CD

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

5.3: Anti-Pattern: Avoid Running the Entire Suite—Split It into Smaller Suites

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

5.4: Anti-Pattern: Prioritizing Tests in Execution

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

5.4.1: A Story of Prioritizing Tests

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

5.4.2: Why Prioritizing Automated Tests is Wrong?

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

5.5: Anti-Pattern: Fake Automated Acceptance Testing

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

5.6: Anti-Pattern: E2E Test Scripts are Not in an Independent Git Repository

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

5.7: Anti-Pattern: Headless Browser Testing

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

5.7.1: The Embarrassing History of Headless Testing

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

5.7.2: Why Fake Automated Testers Love 'Headless Browser Testing'?

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

5.7.3: Why Go for Headless? Mainly speed. Let's talk about Speed.

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

5.7.4: Speed Gain in Individual Test vs Overall Suite

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

5.7.5: Problems with Headless Browser Testing

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

5.8: Anti-Pattern: Long execution time > 1 hour

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

5.9: Anti-Pattern: Parallel Execution on one machine

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

5.9.1: Unaware of Challenges of Running an Automated Test Suite in Parallel

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

5.9.2: Simple Solution: Run one test at a time per machine

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

5.10: Wrap-up

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

6: CT Features Missing in Most CI/CD Servers

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

6.1: View Test Scripts in the CT Server Web Interface

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

6.2: Capture Application Screenshots on E2E Test Failure

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

6.3: Quick navigation to the failed test line

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

6.4: Auto-Retry Failed E2E Test on Another Build Agent

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

6.5: Manually Rerun Tests

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

6.6: Dynamic Test Execution Ordering

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

6.7: Distribution rules

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

6.8: Delay completion

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

6.9: Anti-Pattern: Being constrained by a vendor's CI/CD server limitations and do nothing

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

6.10: Wrap-up

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

7: Mindset

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

7.1: Anti-Pattern: “Test Automation in CI Slows Down Development”

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

7.2: Anti-Pattern: “Prefer expensive commercial CI/CD solutions over free open-source ones”

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

7.3: Anti-Pattern: “API Testing as E2E Test Automation in CI/CD is Enough”

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

7.4: Anti-Pattern: “E2E Test Automation Is Inherently Fragile, Keep Them To a Minimum”

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

7.5: Anti-Pattern: “E2E Test Automation Is Too Expensive, So we don’t include in CI/CD”

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

7.6: Anti-Pattern: Assuming Automated Tests Will Always Pass After Creation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

7.7: Anti-Pattern: “E2E Test Automation Scripts Must Use the Same Language as the Application Code”

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

7.7.1: Which language is best suited for E2E test automation? Ruby.

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

7.8: Anti-Pattern: “Set up a CI/CD Solution Takes a Long Time and Costly”

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

7.9: Anti-Pattern: Drawn to flashy, smoke-and-mirror features like Test Replay

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

7.9.1: Test Replay

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

7.9.1.1: “Test Replay” is good for demonstration, but not useful in real E2E Test Automation.

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

7.9.1.2: Expensive in terms of infrastructure

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

7.9.1.3: Mostly, a waste of time

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

7.10: Wrap-up

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

8: General Practices

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

8.1: Anti-Pattern: Not the exact the same release package for different target environments.

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

8.2: Anti-Pattern: Underinvesting in CI/CD Hardware

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

8.3: Anti-Pattern: Automating the Tear Down and Setup of the CI Servers Daily

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

8.4: Anti-Pattern: No Admin Access for Developers and Testers in CI/CD

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

8.5: Anti-Pattern: Only limited to execute one specific type of automated tests in one language

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

8.6: Anti-Pattern: A Single CI/CD Server Serving the Entire Company/Division

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

8.7: Anti-Pattern: No Daily Builds

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

8.8: Anti-Pattern: No incremental version or build number during internal testing

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

8.9: Anti-Pattern: Commit-Driven CI/CD Triggers

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

8.10: Anti-Pattern: Sending Real Emails Out During Test execution in CI/CD

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

9: Non-Functional Testing

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

9.1: Anti-Pattern: No Continuous Performance Testing

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

9.1.1: Running Performance Tests in the CI/CD Server Makes Perfect Sense

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

9.1.2: Case Study - Performance Testing WhenWise's Database Reset Utility

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

9.2: Anti-Pattern: No Continuous Load Testing

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

9.3: Wrap-up

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

10: CI/CD Hosting

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

10.1: Anti-Pattern: Neglecting the Needs of E2E Test Automation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

10.2: Anti-Pattern: Falling for flashy but useless features in cloud testing platforms but miss the big picture

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

10.3: Anti-Pattern: Completely unaware of or blindly exlcuding on-premise Setup

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

10.3.1: Significant Cost Savings

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

10.3.1.1: Cypress Cloud’s Demise: Primarily Due to Its High Cost

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

10.3.2: Cloud option cost goes up quickly with more test executions, extras.

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

10.3.3: Superior Performance

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

10.3.4: Higher Reliability

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

10.3.4.1: Shared-CPU Plan Lacks Reliability

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

10.3.4.2: Shared-CPU Plan Lacks Consistency

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

10.3.4.3: Use Mac Computers?

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

10.3.4.4: Some considerations drawn from my experience

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

10.3.5: Greater Flexibility

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

10.3.5.1: Supports many testing types

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

10.3.5.2: Supports multiple test automation frameworks in different languages.

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

10.3.5.3: Maximum flexibility

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

10.3.6: Less Security Concerns

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

10.3.6.1: Security Concerns of E2E Test Scripts outside the business

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

10.3.6.2: Access to the Main Test Server Instance(s)

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

10.3.6.3: Access to Supporting Services

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

10.3.7: Easier Implementation Than Most Thought

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

10.4: Wrap-up

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

11: Team Involvement

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

11.1: Execute Individual Tests via the CT Server Web Interface

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

11.2: Data Preparation for User Acceptance Testing

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

11.3: Self-help Utilities that whole team can use on the CT Server

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

11.4: Generate a Traceability Matrix

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

11.4.1: Understand Requirement Traceability first

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

11.4.2: Enable requirement traceability in functional test scripts

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

11.4.3: Generate Requirement Traceability with ease in BuildWise

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

11.5: Anti-Pattern: Non-Tech Team Members Not Using CI/CD Server

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

11.6: Anti-Pattern: Only One Run Configuration

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

11.7: Case Study: A bug-fixing cycle in a team with real CI/CD

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

11.8: Real CI/CD Boots Customers' Trust and Confidence in the team

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

12: Leadership

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

12.1: Anti-Pattern: Failing to Recognize the Great Benefits of Real CI/CD

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

12.2: Anti-Pattern: Lack Team-Wide Training on CI/CD Process

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

12.3: Anti-Pattern: Developers delay fixing newly discovered defects by Continuous Testing

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

12.4: Anti-Pattern: Stuck with One CI Vendor or One Person

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

12.5: Anti-Pattern: Assuming Existing Staff Can Deliver Real CI/CD and E2E Automation with a New Product

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

12.6: Anti-Pattern: Fixated with All-in-one CI/CD. Two Stage CI → CD Often Better

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

Afterword

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

My CI/CD Formula

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

AgileWay Test Automation Formula

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

How many E2E Tests in CI/CD are enough?

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

My Continuous Testing Lab Setup

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

Appendix I: Anti-Patterns List

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

Appendix II: CI Steps

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ci-cd-anti-patterns.

Resources

Books

- **Practical Web Test Automation**[1] by Zhimin Zhan

  Practical Web Test Automation is the book to guide you to the test automation success on web apps using Selenium WebDriver.

- **Practical Continuous Testing**[2] by Zhimin Zhan

  The second book of “Practical” series, focuses on how to effectively execute automated functional tests in a Continuous Testing server.

- **End-to-end Test Automation Anti-Patterns**[3] by Zhimin Zhan.

  Lessons from real-world test automation failures.

Tools

- **TestWise IDE**[4]

  AgileWay’s next-generation functional testing IDE supports Selenium WebDriver and Appium.

- **BuildWise Server**[5]

  AgileWay’s free, open-source and international award-winning Continuous Testing server, purposely designed for running automated UI tests with quick feedback.

[1] https://leanpub.com/practical-web-test-automation
[2] https://leanpub.com/practical-continuous-testing
[3] https://leanpub.com/end-to-end-test-automation-anti-patterns
[4] https://agileway.com.au/testwise
[5] https://agileway.com.au/buildwise

BuildWise Agent: designed to work with the BuildWise server for parallel test execution.

Blog

- **The AgileWay**[6]

  Zhimin and Courtney Zhan share their original insights, lessons, and best practices on Substack, based on more than combined 30 years of hands-on experience in end-to-end test automation, Continuous Testing, and Agile software development.

- **Software Side Hustler**[7]

  In this newsletter, Zhimin draws on more than 20 years of experience to offer insights and practical lessons for ambitious software professionals looking to launch side hustles—whether that involves developing apps, creating digital products like content or books, or generating income through consulting and software testing.

[6] https://agileway.substack.com/
[7] https://softwaresidehustler.substack.com

References

[Beck04] Kent Beck (2004). *Extreme Programming Explained: Embrace Change (2nd ed.)*. Addison-Wesley Progressional.

[Beck02] Kent Beck (2002). *Test Driven Development: By Example*. Addison-Wesley Professional.

[Martin19] Robert C. Martin (2019). *Agile: Back to basics*. Prentice Hall.

[Martin11] Robert C. Martin (2011). *The clean coder: A code of conduct for professional programmers*. Prentice Hall.

[Fowler00] Martin Fowler, "Continuous Integration (original version)" (posted Sep 10, 2010) http://martinfowler.com/articles/originalContinuousIntegration.html

[Poppendieck07] Poppendieck, M., & Poppendieck, T. (2007). Implementing lean software development: From concept to cash. Addison-Wesley Professional.

[McConnell04] McConnell, S. (2004). Code complete (2nd ed.). Microsoft Press.

[CrispinGregory09] Crispin, L., & Gregory, J. (2009). Agile testing: A practical guide for testers and agile teams. Addison-Wesley Professional.

[HumbleFarley10] Humble, J., & Farley, D. (2010). Continuous delivery: Reliable software releases through build, test, and deployment automation. Addison-Wesley Professional.

[ShoreWarden08] Shore, J., & Warden, S. (2008). The art of agile development. O'Reilly Media.

[HuntThomas19] Hunt, A., & Thomas, D. (2019). The pragmatic programmer: Your journey to mastery (20th anniversary ed.). Addison-Wesley Professional.

[SubramaniamHunter06] Subramaniam, V., & Hunt, A. (2006). Practices of an agile developer: Working in the real world. Pragmatic Bookshelf.

About the Cover Photo

Featured on the cover is the iconic Byodoin Template (a World Heritage) in Uji, Japan, photographed by the author in May 2025.

Book Cover Designed by Courtney Zhan.