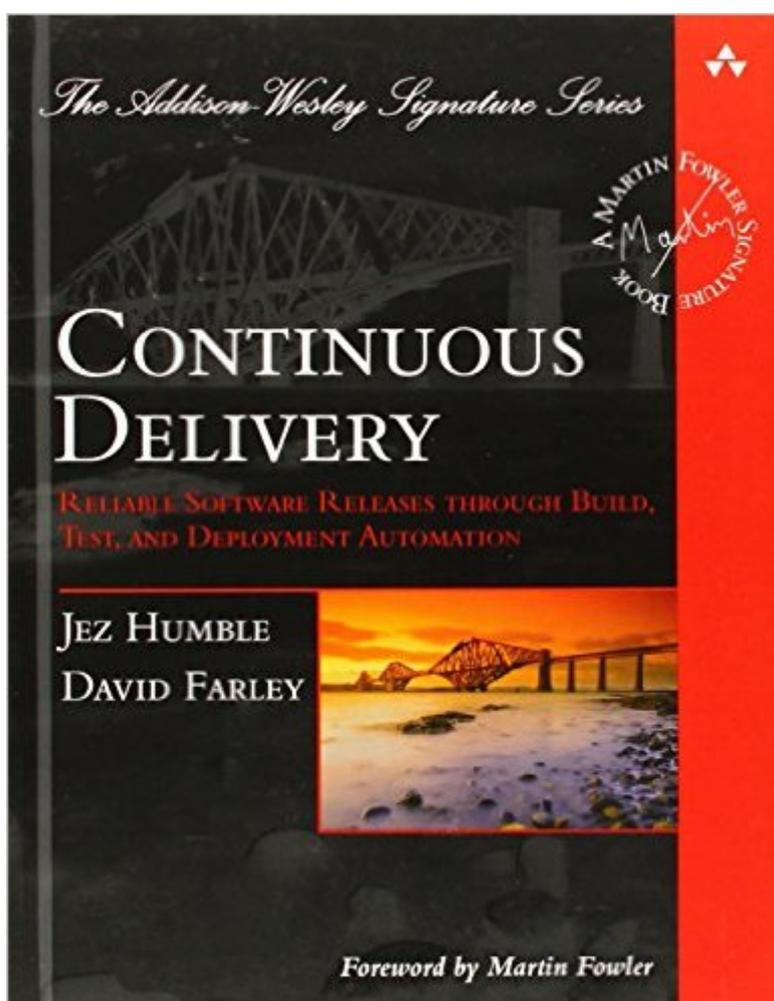# Continuous Delivery: Reliable Software Releases Through Build, Test, And Deployment Automation (Addison-Wesley Signature Series (Fowler))

Synopsis

Winner of the 2011 Jolt Excellence Award!   Getting software released to users is often a painful, risky, and time-consuming process. This groundbreaking new book sets out the principles and technical practices that enable rapid, incremental delivery of high quality, valuable new functionality to users. Through automation of the build, deployment, and testing process, and improved collaboration between developers, testers, and operations, delivery teams can get changes released in a matter of hoursâ " sometimes even minutesâ "no matter what the size of a project or the complexity of its code base. Â Jez Humble and David Farley begin by presenting the foundations of a rapid, reliable, low-risk delivery process. Next, they introduce the â œdeployment pipeline,â • an automated process for managing all changes, from check-in to release. Finally, they discuss the â œecosystemâ • needed to support continuous delivery, from infrastructure, data and configuration management to governance. Â The authors introduce state-of-the-art techniques, including automated infrastructure management and data migration, and the use of virtualization. For each, they review key issues, identify best practices, and demonstrate how to mitigate risks. Coverage includes Â â ¢ Automating all facets of building, integrating, testing, and deploying software â ¢ Implementing deployment pipelines at team and organizational levels â ¢ Improving collaboration between developers, testers, and operations â ¢ Developing features incrementally on large and distributed teams â ¢ Implementing an effective configuration management strategy â ¢ Automating acceptance testing, from analysis to implementation â ¢ Testing capacity and other non-functional requirements â ¢ Implementing continuous deployment and zero-downtime releases â ¢ Managing infrastructure, data, components and dependencies â ¢ Navigating risk management, compliance, and auditing Â Whether youâ ™re a developer, systems administrator, tester, or manager, this book will help your organization move from idea to release faster than everâ "so you can deliver value to your business rapidly and reliably. Â

Book Information

Series: Addison-Wesley Signature Series (Fowler)

Hardcover: 512 pages

Publisher: Addison-Wesley Professional; 1 edition (August 6, 2010)

Language: English

ISBN-10: 0321601912

ISBN-13: 978-0321601919

Product Dimensions: 7.3 x 1.2 x 9.3 inches

Shipping Weight: 2.1 pounds (View shipping rates and policies)

Average Customer Review: 4.5 out of 5 starsÂ Â See all reviewsÂ (84 customer reviews)

Best Sellers Rank: #15,473 in Books (See Top 100 in Books)   #6 inÂ Books > Computers & Technology > Programming > Software Design, Testing & Engineering > Testing   #17 inÂ Books > Textbooks > Computer Science > Software Design & Engineering   #29 inÂ Books > Computers & Technology > Programming > Software Design, Testing & Engineering > Software Development

Customer Reviews

This book is packed full of great ideas, but it suffers from painful redundancy. In response to another review, an author claims that it was intentional, so that one could skip around without reading from cover to cover. My response to that is that they should have had better editors. I have read many technical books designed for skipping around. None were as tediously repetitive as this one. Eventually, one has to expect that the reader is going to read more than one chapter and might even remember something from a previous chapter and do them the courtesy of not belaboring the main points each time. It's not even limited to once per chapter. The repetition frequently continues within each chapter, section by section.That said, there are some good gems inside. My favorite parts might be the many real-world stories of how things can go wrong or how applying some of the principles smoothed things out. The detail, diversity and verisimilitude of those anecdotes sets the book apart from many books in the field.I wish I could say this was a "must have" book, but it's really more of a "must skim" sort of book.

Continuous Delivery from Jezz Humble and David Farley is an important contribution to the field of software development. It takes continuous integration to the logical conclusion and covers how to set up a continuous integration system which covers everything from check-in to delivery to production. It doesn't state you have to deliver directly in production, but it will explain how technically it is achievable to do that and what enormous benefits this brings to your organization.Continuous delivery consists of three parts: 1) Foundation, 2) Deployment Pipeline, and 3) Delivery EcosystemThe first four chapters cover the fundamentals the rest of the book is based on. The first chapter provides some problems with more traditional approaches and also introduces some principles extracted out of continuous delivery. The next three chapters cover topics that provide the basics of continuous delivery. Someone involved with agile development for a while is probably aware of most of this and it will be a quick read. For new people, these chapter provide a quick introduction to these topics so that you can understand the rest of the book. The

chapters are: "configuration management," "continuous integration," and "implementing a testing strategy."The second part is the core of the book. It explains the continuous delivery pipeline. This pipeline is a series of stages (a series of continuous integration systems) each stage covering higher-level wider-range of testing so that the confidence in the product increases the later the stage in the deployment pipeline passes. The stages the authors recommend in the deployment pipeline are: commit, acceptance, capacity, manual, production. Each of these stages (except for manual) has its own chapter which explains tools and practices that the authors have found useful in that stage of the deployment pipeline. The part also contains an additional 'foundation' chapter about build and deployment scripting.The last part of the book is one that I myself found most interesting which covers perhaps some 'advanced' topics. The part is called "delivery ecosystem" and the chapters aren't directly related to each other but each chapter covers a common topic related to the deployment pipeline. Chapter 11 talks about managing and automating your infrastructure as part of your build also. It introduces a vast amount of topics related to automation (pupper, chef), virtualization, cloud computing and monitoring. Unfortunately, the book is only able to touch a little upon each of these topics as each of them could easily fill several others books (and they do!). Chapter 12 covers a very frequent problem in testing and test automation related to managing data. It explains several different approaches and then evaluates them and shares the experiences and recommendations of the authors. Managing test data is a common problem and is rarely covered in the amount of detail as this book does. Chapter 13 discusses different scaling options by componentizing the product and what effect this has on the continuous deployment pipeline (basically adding another dimension to the pipeline). Chapter 14 is about version control and can be summarized as "avoid branching" but the authors do a good job explaining that message and why the alternatives are indeed worst. Chapter 15 was a short (and I slightly disliked this chapter) about managing continuous delivery. It felt like the standard "and now... what actions to take"-chapter. It was a bit shallow though.When the book was published, I read it through rather quickly and liked it but didn't appreciated the depth of the book yet. I re-read it the second time more thoroughly and enjoyed the careful comparisons and explanations of the recommendations of the authors. They shared the experiences they have had very clear. The book is interesting to me as it covers a vast area and thus it is hard to not touch everything shallowly, but they don't, they go in more depth at the points where the authors feel it is appropriate (for example, parts that are controversial or often done differently).The book isn't perfect though! As some other reviewers pointed out, it is repetitive and should have been thinner. I agree with that. Also, sometimes the book side-tracks in interesting facts that are unlikely to help the reader a lot such as the history of version control. Next, the book

contains some very basic things that could have perhaps been left out (or put as appendix), such as an explanation of maven. My last comment is that the book sometimes contradicts itself such as the recommendation to do things "at the beginning of the project" but then later stating that "at the beginning of the project, all these decisions will change". There I still felt the influence of standard 'project' thinking.With all these drawbacks, I still decided to rate the book five stars because I do think it is an very influential and important book. It tells and *shows* that continuous delivery is not just a perfection state but that it can be achieved today. Not only that, it can be achieved in larger projects, not just small web projects. This is a huge contribution to the industry and I think and hope that the practices of continuous delivery will become standard practices everywhere. Excellent read (except for the repetition) and highly recommended.

This is one of the most important software books published in years. From the beginning and throughout the book, the authors emphasize the importance in establishing one delivery team consisting of various experts throughout the software lifecycle - developers, DBAs, Systems/Operations, network specialists, testers and so on. The overarching pattern the authors describe is the Deployment Pipeline, which is basically a staged process consisting of all of the steps to go from bare/virtual metal to a working system whenever there is a change to source files. Of course, the only way this can be done is through copious amounts of automation. The other key point the authors make is that this automated delivery system - itself - is versioned with every change. Not just the custom source code, but also the operating system(s), tools, configuration and everything necessary to create a working software system - a crucial aspect of the Deployment Pipeline.To sum up key points from the book in a few bullets:* The purpose of Continuous Delivery is to reduce the cycle time between an idea and usable software* Automate (almost) everything necessary to create usable software* Version complete software systems (not just source code) for every change committed to version control system* Employ a Deployment Pipeline in which the entire system is recreated whenever a change is committed to the version-control system and provide continuous feedback* Identify one delivery team consisting of various delivery experts - build, deploy, provisioning, database, testing, etc. - a concept emphasized in the DevOps movementThe authors go into great detail in describing each of these themes. So, if you want the process of delivering software to any target environment - including production - to be a click of a button and something that can be accomplished as often as the business requires, get this book. When you employ the practices in this book, no longer will you need to artificially throttle changes delivered to users for months or even years because of the expense and risk required to deliver

software.

Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation (Addison-Wesley Signature Series (Fowler)) Developer Testing: Building Quality into Software (Addison-Wesley Signature Series (Cohn)) Coaching Agile Teams: A Companion for ScrumMasters, Agile Coaches, and Project Managers in Transition (Addison-Wesley Signature Series (Cohn)) The Scrum Field Guide: Agile Advice for Your First Year and Beyond (2nd Edition) (Addison-Wesley Signature Series (Cohn)) Large-Scale Scrum: More with LeSS (Addison-Wesley Signature Series (Cohn)) Agile Product Management with Scrum: Creating Products that Customers Love (Addison-Wesley Signature Series (Cohn)) Essential Scrum: A Practical Guide to the Most Popular Agile Process (Addison-Wesley Signature Series (Cohn)) Agile Product Management with Scrum: Creating Products that Customers Love (Adobe Reader) (Addison-Wesley Signature Series (Cohn)) Management 3.0: Leading Agile Developers, Developing Agile Leaders (Adobe Reader) (Addison-Wesley Signature Series (Cohn)) Management 3.0: Leading Agile Developers, Developing Agile Leaders (Addison-Wesley Signature Series (Cohn)) The Lean Mindset: Ask the Right Questions (Addison-Wesley Signature Series (Beck)) Deployment Fundamentals, Vol. 6: Deploying Windows 10 Using Microsoft Deployment Toolkit REI Automation Blueprint The A-Z Blueprint To Automate Your Real Estate Business: REI Automation Blueprint The A-Z Blueprint To Automate Your Real ... Brittney Calloway of Top Notch Consulting Continuous Delivery and DevOps: A Quickstart Guide - Second Edition Agile IT Organization Design: For Digital Transformation and Continuous Delivery Continuous Delivery with Visual Studio ALM 2015 Fetal Growth and the Placenta-From Implantation to Delivery: From Implantation to Delivery (Trophoblast Research) Formal Languages and Their Relation to Automata (Addison-Wesley Series in Computer Science and Information Processing) Data Visualization Toolkit: Using JavaScript, Rails, and Postgres to Present Data and Geospatial Information (Addison-Wesley Professional Ruby Series) The Design and Implementation of the 4.3 Bsd Unix Operating System: Answer Book (Addison-Wesley series in computer science)