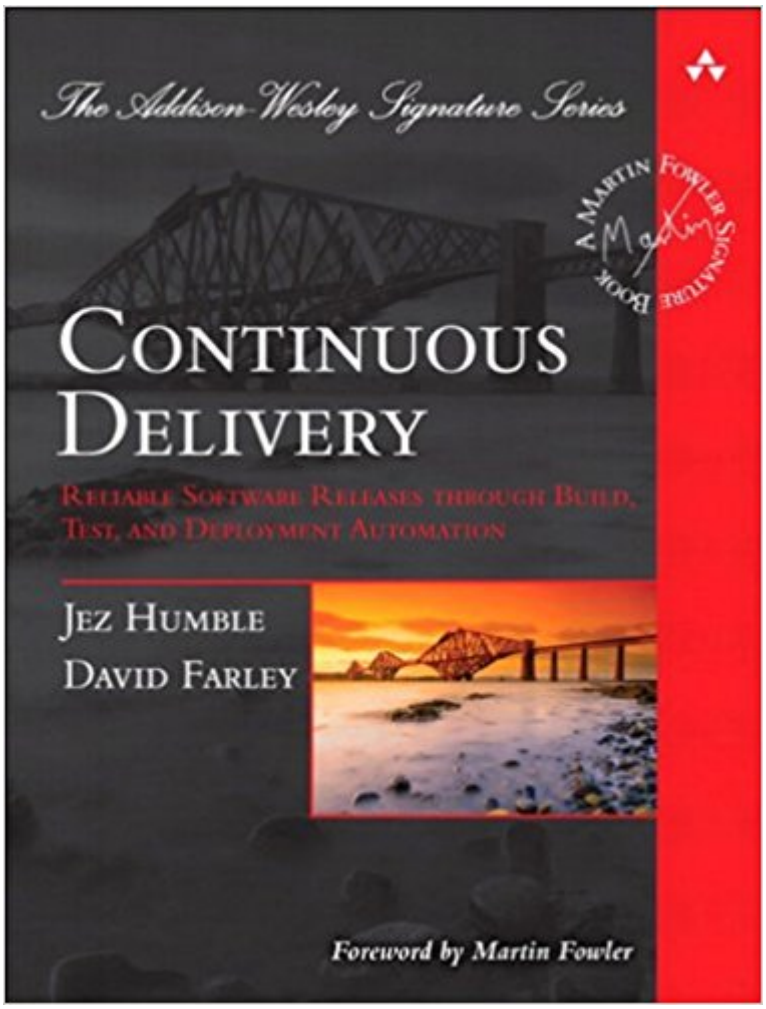
Continuous Delivery: Reliable Software Releases Through Build, Test, And Deployment Automation (Adobe Reader) (Addison-Wesley Signature Series (Fowler))



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

File Size: 8290 KB

Print Length: 501 pages

Page Numbers Source ISBN: 0321601912

Simultaneous Device Usage: Up to 5 simultaneous devices, per publisher limits

Publisher: Addison-Wesley Professional; 1 edition (July 27, 2010)

Publication Date: July 27, 2010

Sold by:Â Digital Services LLC

Language: English

ASIN: B003YMNVC0

Text-to-Speech: Enabled

X-Ray: Enabled

Word Wise: Enabled

Lending: Not Enabled

Enhanced Typesetting: Enabled

Best Sellers Rank: #33,838 Paid in Kindle Store (See Top 100 Paid in Kindle Store) #8 inÂ Books > Computers & Technology > Programming > Software Design, Testing & Engineering > Testing #15 inÂ Kindle Store > Kindle eBooks > Computers & Technology > Programming > Software Design > Software Development #49 inÂ Books > Computers & Technology > Programming > Software Design, Testing & Engineering > Software Development

Customer Reviews

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.

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.

(Bradygames Signature Guides) Automation Made Easy: Everything You Wanted to Know about Automation--and Need to Ask The Design and Implementation of the 4.4 BSD Operating System (Addison-Wesley UNIX and Open Systems Series) R for Everyone: Advanced Analytics and Graphics (Addison-Wesley Data & Analytics Series) First Principles of Discrete Systems and Digital Signal Processing (Addison-Wesley Series in Electrical Engineering) Apache Hadoop YARN: Moving beyond MapReduce and Batch Processing with Apache Hadoop 2 (Addison-Wesley Data & Analytics Series) Principles of Compiler Design (Addison-Wesley series in computer science and information processing) Circuits, Interconnections, and Packaging for Vlsi (Addison-Wesley VLSI systems series) Hadoop 2 Quick-Start Guide: Learn the Essentials of Big Data Computing in the Apache Hadoop 2 Ecosystem (Addison-Wesley Data & Analytics Series) TCP/IP Illustrated, Vol. 1: The Protocols (Addison-Wesley Professional Computing Series)