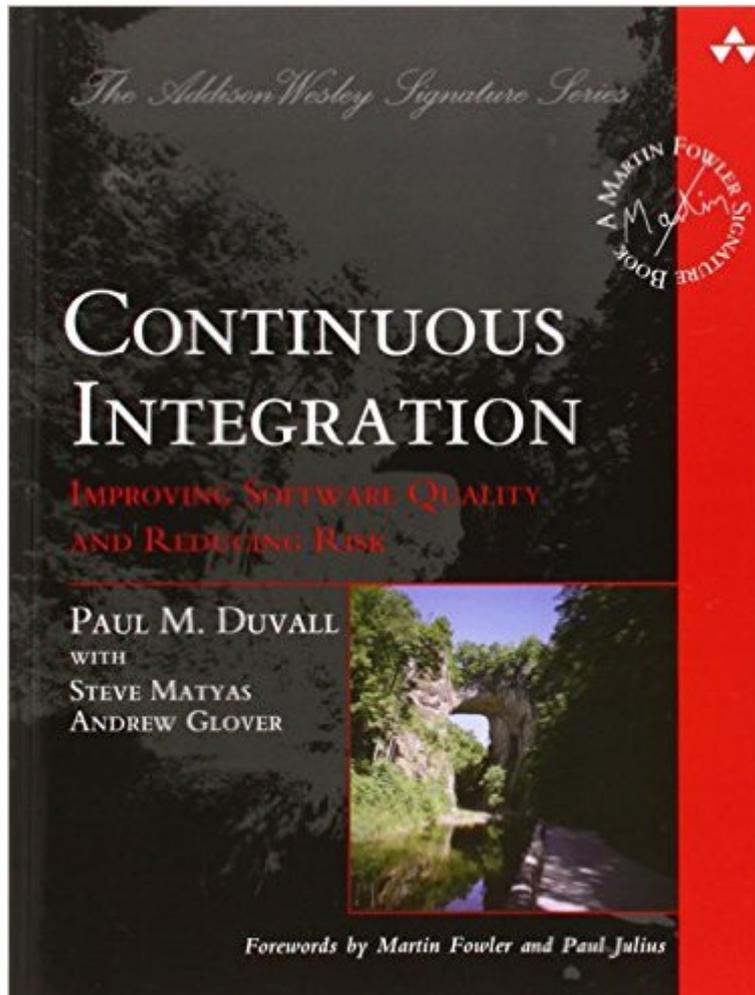# Continuous Integration: Improving Software Quality And Reducing Risk

Synopsis

For any software developer who has spent days in â œintegration hell,â • cobbling together myriad software components, Continuous Integration: Improving Software Quality and Reducing Risk illustrates how to transform integration from a necessary evil into an everyday part of the development process. The key, as the authors show, is to integrate regularly and often using continuous integration (CI) practices and techniques. Â The authors first examine the concept of CI and its practices from the ground up and then move on to explore other effective processes performed by CI systems, such as database integration, testing, inspection, deployment, and feedback. Through more than forty CI-related practices using application examples in different languages, readers learn that CI leads to more rapid software development, produces deployable software at every step in the development lifecycle, and reduces the time between defect introduction and detection, saving time and lowering costs. With successful implementation of CI, developers reduce risks and repetitive manual processes, and teams receive better project visibility. Â The book covers How to make integration a â œnon-eventâ • on your software development projects How to reduce the amount of repetitive processes you perform when building your software Practices and techniques for using CI effectively with your teams Reducing the risks of late defect discovery, low-quality software, lack of visibility, and lack of deployable software Assessments of different CI servers and related tools on the market The bookâ ™s companion Web site, www.integratebutton.com, provides updates and code examples. Â

Book Information

Paperback: 336 pages

Publisher: Addison-Wesley Professional; 1 edition (July 9, 2007)

Language: English

ISBN-10: 9780321336385

ISBN-13: 978-0321336385

ASIN: 0321336380

Product Dimensions: 7 x 0.9 x 9.1 inches

Shipping Weight: 1.1 pounds (View shipping rates and policies)

Average Customer Review: 4.4 out of 5 starsÂ Â See all reviewsÂ (28 customer reviews)

Best Sellers Rank: #131,857 in Books (See Top 100 in Books)   #1 inÂ Books > Computers & Technology > Programming > Software Design, Testing & Engineering > Quality Control   #51 inÂ Books > Computers & Technology > Programming > Software Design, Testing & Engineering >

Customer Reviews

Continuous Integration refers to the practice of automating the build, testing and deployment of your software, so that producing a finished executable (and the related artifacts) can be done at the touch of a button, and is ideally carried out several times a day.If this seems like a nice to have feature of your own development, but less of a core practice when compared to version control and comprehensive tests (both of which are requirements for doing CI), this book does a pretty good job of advocating CI as being just as important.First, the book introduces the core practices of CI (regular builds, tests, and deployment), then goes on to demonstrate how it facilitates other, more advanced practices, which gain value when automated, such as enforcing code style, and recording code metrics.It does not assume any particular platform, although most of the code uses Java and C# (and associated XML configuration). As a result, it will appeal most to those who want general guidance about why CI is a good idea, what to put under CI, how often to integrate, how long to allow the build to take, what to do if builds are too slow, etc. There's clearly no one-size-fits-all answer to this and this not a step-by-step tutorial book, so you'll need to adapt the code samples given in this book for your own ends.Therefore, if you're completely new to the idea of CI, then maybe you might want to check out Mike Clark'sÂ Pragmatic Project AutomationÂ first, which covers a lot of the same ground as the first part of this book, but goes into a lot more detail about the mechanics of using Ant and JUnit with Cruise Control.Apart from being more language agnostic, what takes this book beyond the Pragmatic tome is the second part, which demonstrates the more advanced processes that CI makes possible: including a chapter on how to integrate databases into CI, which touches on some cultural issues (e.g. the DBA being separate from the rest of the coding team) and providing sandboxes for each developer. Additionally, there's material on how to include reporting and analysis, e.g. code duplication, code coverage and static analysis tools such as Java's FindBugs.It's also a quick and easy read (less than 300 pages), while still having a pretty wide purview. I don't think this is a subject that would benefit from an enormous tome, and you'll still come away with a much clearer idea of your project's automation and scheduling needs, although you might have to do a bit of digging in online documentation of the various tools mentioned in the book to find your exact solution.The only bad thing I have to say about this book is that there are some very brief developer dialogues sprinkled throughout, used as examples to highlight suboptimal practices. As ever, these are cringe-inducing and artificial.Out of the core agile practices of unit testing, version control, and project automation, the latter has the least amount of material available

to read. Fortunately, this is a readable, persuasive and helpful book for curing the big bang integration blues.

This book is an excellent overview of why Continuous Integration is important and about more than just compiling frequently. The book helps you to understand why to do CI, what you can do beyond building, and how to do it. In addition to general principles, the book points you to some excellent tools and resources. This book is an excellent companion to Software Configuration Management Patterns: Effective Teamwork, Practical Integration; it provides teriffic information that support the build patterns in that book. You might already know some of the information in this book, but it is worth buying if you need to encourge CI in your organization for the clear discussion of why CI matters and the for the detailed advice on how to implement it.

This book is a must read for everyone delving into the practice of Continuous Integration. In my opinion, Continuous Integration is one of the most important practices to really achieve agility and to mitigate risks related to architecture and software integration.The book contains more than 40 practices related to this important subject. For me, an experienced software engineer who already uses and knows a lot of CI tools, the best chapters are those which illuminate how to do Continuous Database Integration (Chapter 5), Continuous Testing (Chapter 6) and Continuous Inspection (Chapter 7).Another great plus of this book is Appendix B on how to evaluate CI Tools. It gives a lot of hints to choose the right CI tool for your project or company.Even if you are an experienced CI practitioner this book is a welcome addition. It shows why each practice is important and what are the benefits to use it on a SW development project.If you are a beginner or intermediate practitioner in the Continuous Integration World this book is a must have. You will receive a lot of wisdom collected by the authors during their careers.If you don't know what is Continuous Integration read the great article by Martin Fowler.Spare you a lot of integration and software quality headaches reading this excellent book!

"Continuous Integration" is part of Addison Wesley series. This series includes books like "Refactoring to Patterns." "Continuous Integration" definitely meets the standards of this series.Each chapter describes CI related practices. There is a chapter dedicated to risks reduced by CI including anti-patterns like "It works on my machine." Each chapter ends with questions to get you thinking about CI in YOUR process. I particularly like how the authors address the "CI is good by my project is special" problem.The authors give examples in different languages including Java, .NET and

Ruby. The appendices on resources and tools are very useful. The book goes beyond CI and addresses continuous inspection and deployment. My only problem when reading the book is that I forgot I was supposed to be writing a review. It was so good, I just got caught up in the book!Do check out the companion website - integratebutton.com. It currently contains video examples of three practices described in the book. The materials are presented in slide and diagram format. It reinforced the book nicely because it was like a guru explaining his experiences. It also goes into much more detail than the book has room for on each topic.This is an excellent book and the website adds to it!* For the JavaRanchers reading this, the first sentence in the book is a quote from Kathy Sierra.

Continuous Integration: Improving Software Quality and Reducing Risk Lean Six Sigma: The Ultimate Guide To Lean Six Sigma With Tools For Improving Quality And Speed! (Lean, Six Sigma, Quality Control) Practical CGM: Improving Patient Outcomes through Continuous Glucose Monitoring M&A Integration: How To Do It. Planning and delivering M&A integration for business success Mclaughlin And Kaluzny's Continuous Quality Improvement In Health Care Managing Software Quality and Business Risk (Rights of Children) Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation (Addison-Wesley Signature Series (Fowler)) Agile Software Engineering with Visual Studio: From Concept to Continuous Feedback (Microsoft Windows Development Series) Agile Software Engineering with Visual Studio: From Concept to Continuous Feedback (2nd Edition) (Microsoft Windows Development Series) The Feeling of Risk: New Perspectives on Risk Perception (Earthscan Risk in Society) Software Quality Assurance: In Large Scale and Complex Software-intensive Systems Software Process Design: Out of the Tar Pit (Mcgraw-Hill International Software Quality Assurance) The Lean Six Sigma Pocket Toolbook: A Quick Reference Guide to 100 Tools for Improving Quality and Speed The Lean Six Sigma Pocket Toolbook: A Quick Reference Guide to Nearly 100 Tools for Improving Quality and Speed Lean Hospitals: Improving Quality, Patient Safety, and Employee Engagement, Third Edition The Quality Solution: The Stakeholder's Guide to Improving Health Care Surreptitious Software: Obfuscation, Watermarking, and Tamperproofing for Software Protection: Obfuscation, Watermarking, and Tamperproofing for Software Protection Software Engineering Classics: Software Project Survival Guide/ Debugging the Development Process/ Dynamics of Software Development (Programming/General) Axiomatic Quality: Integrating Axiomatic Design with Six-Sigma, Reliability, and Quality Engineering Quality Management Exam Review for Radiologic Imaging Sciences (Quality Management Review)