

<<深入浅出软件开发>>

图书基本信息

书名：<<深入浅出软件开发>>

13位ISBN编号：9787564112363

10位ISBN编号：7564112360

出版时间：2009-1

出版时间：东南大学出版社

作者：（美）皮隆（Pilone, D.），（美）迈尔斯（Miles, R.）著

页数：459

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

<<深入浅出软件开发>>

内容概要

想知道何为测试驱动开发吗？

想知道高级顾问在按小时计费的咨询中是如何应用最佳实践的吗？

或者也许您正准备自动构建您的软件，为代码使用版本控制，重构以及在软件中集成一些设计模式。当您读完这本《深入浅出软件开发》的时候，您将能够跟踪项目进度，统计团队每个部分的速度以及反复进行需求分析、设计、开发和部署。

我们认为您用来学习新概念的时间是极其宝贵的。

《深入浅出软件开发》使用认知科学和学习理论的最新研究成果，为您带来多感官的学习体验，它采用了丰富、可视、适于大脑工作的形式，而不是用过多的文字使您感到昏昏欲睡。

本书为英文影印版。

书籍目录

Intro1 great software development: Pleasing your customer Tom's Trails is going online Most projects have two major concerns The Big Bang approach to development Flash forward: two weeks later Big bang development usually ends up in a big MESS Great software development is... Getting to the goal with ITERATION Each iteration is a mini-project Each iteration is QUALITY software The customer WILL change things up It's up to you to make adjustments But there are some BIG problems... Iteration handles change automatically (well sort of) Your software isn't complete until it's been RELEASED Tools for your Software Development Toolbox2 gathering requirements: Knowing what the customer wants Orion's Orbits is modernizing Talk to your customer to get MORE information Bluesky with your customer Sometimes your bluesky session looks like this... Find out what people REALLY do Your requirements must be CUSTOMER-oriented Develop your requirements with customer feedback User stories define the WHAT of your project... estimates define the WHEN Cubicle conversation Playing Planning Poker Put assumptions on trial for their lives A BIG user stoW estimate is a BAD user story estimate The goal is convergence The requirement to estimate iteration cycle Finally, we're ready' to estimate the whole project3 project planning: Planning for success Customers want their software NOW! Prioritize with the customer We know what's in Milestone 1.0 (well, maybe) If the features don't fit, re-prioritize More people sometimes means diminishing returns Work your way to a reasonable milestone 1.0 Iterations should be short and sweet Comparing your plan to reality Velocity accounts for overhead in your estimates Programmers think in UTOPIAN days... Developers think in REAL-WORLD days... When is your iteration too long? Deal with velocity BEFORE you break into iterations Time to make an evaluation Managing customers The Big Board on your wall How to ruin your team's lives4 user stories and tasks: Getting to the real work Introducing iSwoon Do your tasks add up? Plot just the work you have left Add your tasks to your board Start working on your tasks A task is only in progress when it's IN PROGRESS What if I'm working on two things at once? Your first standup meeting... Task 1 : Create the Date class Standup meeting: Day 5, end of Week 1... Standup meeting: Day 2, Week 2... We interrupt this chapter... You have to track unplanned tasks Unexpected tasks raise your burn-dovm rate Velocity helps, but... We have a lot to do... ...but we know EXAC, TLY where we stand Velocity Exposed5 good-enough design: Getting it done with great design iSwoon is in serious trouble... This design breaks the single responsibility principle Spotting multiple responsibilities in your design Going from multiple responsibilities to a single responsibility Your design should obey the SRP, but also be DRY... The post-refactoring standup meeting... Unplanned tasks are still just tasks Part of your task is the demo itself When everything's complete, the iteration's done 6 version control: Defensive development You've got a new contract BeatBox Pro And now the GUI work... Demo the new BeatBox for the customer Let's start with VERSION CONTROL First set up your project... ...then you can check code in and out. Most version control tools will try and solve problems for you The server tries to MERGE your changes If your software can't merge the changes, it issues a conflict More iterations, more stories... We have more than one version of our software... Good commit messages make finding older software easier Now you can check out Version 1.0 (Emergency) standup meeting Tag your versions Tags, branches, and trunks, oh my !

Fixing Version 1.0...tbr real this time. We have TWO code bases now When NOT to hranch... The Zen of good branching What version control does... Version control can't make sure you code actually works... Tools for your Software Development Toolbox6.5 building your code: Insert tab a into slot b... Developers aren't mind readers Building your project in one step Ant: a build tool forJava projects Projects, properties, targets, tasks Good build scripts... Good build scripts go BEYOND the basics Your build script is code, too New developer, take two Tools for your Software Development Toolbox7 testing and continuous integration: Things fall apart Things will ALWAYS go wrong... There are three ways to look at your system... Black-box testing focuses on INPUT and OUTPUT Grey-box testing gets you CLOSER to the code White-box testing uses inside knowledge Testing EVERYTHING with one step Automate your tests with a testing framework Use your framework to run your tests At the wheel of CI with CruiseControl Testing guarantees things will work..., right? Testing all your code means

<<深入浅出软件开发>>

testing EVERY BRANCH Use a coverage report to see what's covered Getting good coverage isn't always easy... What CM does... Tools for your Software Development Toolbox8 test-driven development: Holding your code accountable Test FIRST, not last So we're going to test FIRST... Welcome to test-driven development Your first test... ...fails miserably. Get your tests to GREEN Red, green, refactor... In TDD, tests DRIVE your implementation Completing a task means you've got all the tests you need, and they all pass When your tests pass, move on! Simplicity means avoiding dependencies Always write testable code When things get hard to test, examine your design The strategy pattern provides for multiple implementations of a single interface Keep your test code with your tests Testing produces better code More tests always means lots more code Strategy patterns, loose couplings, object stand ins... We need lots of different, but similar, objects What if we generated objects? A mock object stands in for real objects- Mock objects are working object stand-ins Good software is testable... It's not easy bein' green... A day in the life of a test-driven developer-.. Tools for your Software Development Toolbox9 ending an iteration: It's all coming together... Your iteration is just about complete... ...but there's lots left you could do System testing MUST be done... ...but WHO does system testing? System testing depends on a complete system to test Good system testing requires TWO iteration cycles More iterations means more problems Top 10 Traits of Effective System Testing The life (and death) of a bug So you found a bug.... Anatomy of a bug report But there's still plenty left you COULD do... Time for the iteration review Some iteration review questions A GENERAL priority list for getting EXTRA things done Tools for your Software Development Toolbox10 the next iteration:/f it ain't broke...you still better fix it What is working software? You need to plan for the next iteration Velocity accounts for..., the REAL WORLD And it's STILL about the customer Someone else's software is STILL just software Customer approval? Check! Testing your code Houston, we really do have a problem... Trust NO ONE It doesn't matter who wrote the code. If it's in YOUR software, it's YOUR responsibility. You without your process You with your process11 bugs: Squashing bugs like a pro Previously on Iteration 2 First, you've got to talk to the customer Priority one: get things buildable We could fix code... ...but we need to fix functionality Figure out what functionality works NOW you know what's not working What would you do? Spike test to estimate What do the spike test results tell you? Your team's gut feel matters Give your customer the bug fix estimate Things are looking good... ...and you finish the iteration successfully! AND the customer is happy Tools tbr your Software Development Toolbox12 the real world: Having a process in life Pinning down a software development process A good process delivers good software Formal attire required... Some additional resources... More knowledge == better process Tools for your Software Development ToolboxAppendix 1 leftoversAppendix 2 techniques and principles

<<深入浅出软件开发>>

章节摘录

插图：

<<深入浅出软件开发>>

编辑推荐

“《深入浅出软件开发(影印版)》是本汇集了奇思妙想的书籍，经过精心地设计，它利用一系列图表将信息准确而又清晰全面地映入您的大脑。

它是一本全新类型的书籍。

”——Scott Hanselman，软件开发者、演说家、作家Computer Zen “《深入浅出软件开发(影印版)》讲述了软件开发中那些极少在课堂上出现但却亟需您了解的内容。

”——Keith Wichmann，SOA架构师，Johns Hopkins University Applied Physics Laboratory “无论您在软件开发这一行工作了多久，《深入浅出软件开发(影印版)》都将带给您用于成功开发整个项目的必备工具。

”——Adam Z Szvmanski，软件项目经理Naval Research Laboratory

<<深入浅出软件开发>>

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>