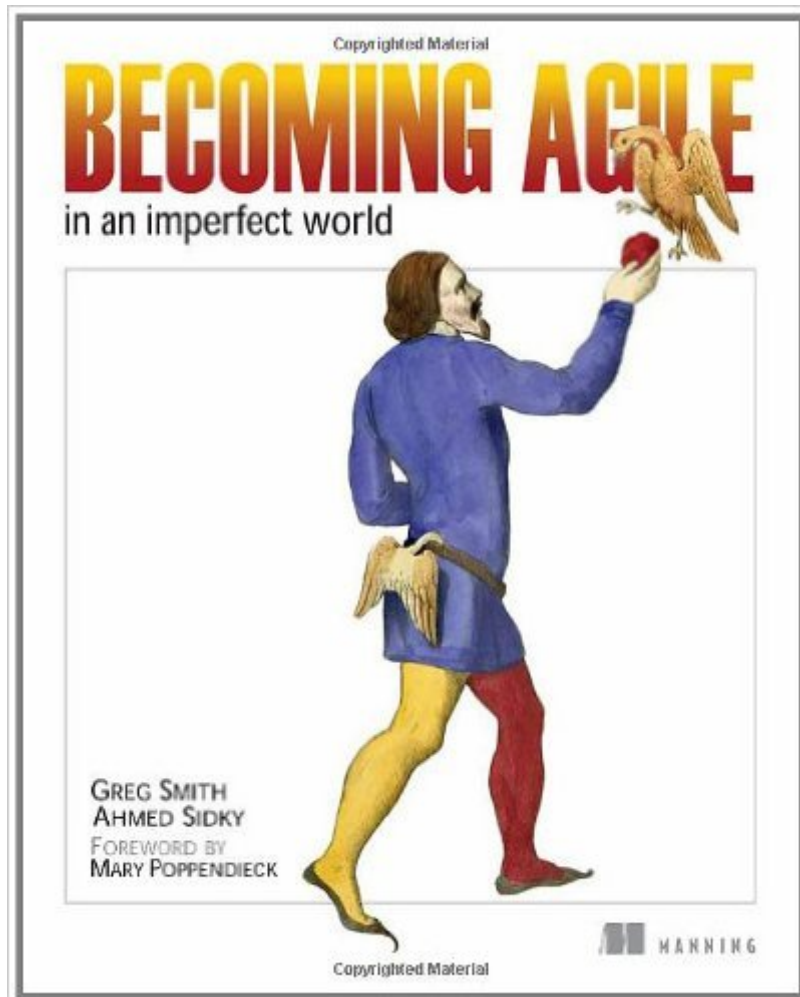# Becoming Agile: ...in An Imperfect World

Synopsis

Many books discuss Agile from a theoretical or academic perspective. Becoming Agile takes a different approach and focuses on explaining Agile from a case-study perspective. Agile principles are discussed, explained, and then demonstrated in the context of a case study that flows throughout the book. The case study is based on a mixture of the author's real-world experiences. Becoming Agile also focuses on the importance of adapting Agile principles to the realities of your environment. In the early days of Agile, there was a general belief that Agile had to be used in all phases of a project, and that it had to be used in its purest form. Over the last few years, reputable Agile authorities have begun questioning this belief: We're finding that the best deployments of Agile are customized to the realities of a given company. Becoming Agile discusses the cultural realities of deploying Agile and how to deal with the needs of executives, managers, and the development team during migration. The author discusses employee motivation and establishing incentives that reward support of Agile techniques.

Book Information

Paperback: 408 pages

Publisher: Manning Publications; 1 edition (June 18, 2009)

Language: English

ISBN-10: 1933988258

ISBN-13: 978-1933988252

Product Dimensions: 7.4 x 0.9 x 9.2 inches

Shipping Weight: 1.5 pounds (View shipping rates and policies)

Average Customer Review: 4.6 out of 5 starsÂ Â See all reviewsÂ (15 customer reviews)

Best Sellers Rank: #672,956 in Books (See Top 100 in Books)   #114 inÂ Books > Business & Money > Management & Leadership > Project Management > PMP Exam   #319 inÂ Books > Computers & Technology > Graphics & Design > User Experience & Usability   #893 inÂ Books > Textbooks > Computer Science > Software Design & Engineering

Customer Reviews

Becoming Agile by Greg Smith and Ahmed Sidky describes how to start your journey to start agile development. The target audience for this book are people who are just starting to look at Agile development and currently have issues. However, personally, I would not recommend this book. The book contains recommendations that I would personally not follow (I'm not that they wouldn't work) and a better alternative might be Mike Cohns new book "Succeeding with Agile" which will be

published later this year.The book contains eight parts and it follows one case study of "Acme Media" throughout the book. Acme media is implementing a small pilot project using agile development. The first chapter of the book describes agile development, the agile manifesto and introduces the background of the case study. The second chapter in the book covers the start, that is the preparation work that the authors recommend. Chapter 4 in this part discusses a assessment test for assessing your agility. The authors recommend to use the test and based on that decide what practices to start with. Chapter 6 describes one of the their key recommendations, which is to form a "core team" which defines the agile process to be used by the organization.Part three describes the real kick-off of the development. Chapter 10 covers a feasibility phase where the a smaller team will assess if the project is even feasible and after that will be a "gate" or a go/no-go moment so that the project can be killed early. It suggests creating a feasibility study guide to do your feasibility and the authors provide some examples. Chapter 11 talks about creating the initial agile pilot team.Part four is called "populating the product backlog" where the authors start describing their variant on user stories called "feature cards." It wasn't clear to me why the authors decided to re-invent or rename other agile practices. The chapter describes creating the feature cards and putting them in the product backlog. Chapter 13 discusses the prioritization and chapter 14 the estimation of the features (which seems to be the wrong order to me, but the authors insist on not wasting estimation time). The chapter explains planning poker and story point (wonder why they are not called feature points...)Part five then discusses the planning (or scheduling) of the project and fills all the requirements to the iterations. Chapter sixteen then starts with the iteration planning of the first iteration. The authors suggest first some modeling and than task breakdown and already doing task assignments. They not that people often recommend differently, but then they assume that it because in these teams "everyone can do everything." It feels like the authors think it is either pre-assigning all tasks OR everyone can do everything... and there is no middle way.Part six discusses the iteration itself and covers some of the agile engineering practices, but only on a somewhat shallow level. The part starts with chapter 17 which still recommends to run an iteration 0 (even though their project has only 2 iterations!!) Part seven discusses change and adaptation. Chapter 20 describes the adaptations of Acme and why they did it. Interestingly enough, the authors suggest a "adapt week" between their iterations. Then chapter 21 describes the final delivery.Part eight, the final part, describes how the core group can take the result of the pilot and try to roll out agile in the whole company.Why I wouldn't recommend the book? There are two main reasons for this. The first is that the book doesn't feel well researched at all and the language is somewhat too 'popular.' For example: page 6 about the Agile manifesto talks about "a group of authors writing a

document." Unless you interpret 'authors' very broadly, this is just not true. On page 33 the authors describe weaknesses of Scrum. I'm probably not the most neutral person about this, but still the sentence "Scrum doesn't want specialists" seems like a really odd conclusion and seems to me simply untrue. Another example, on page 248 the authors equate exploratory testing to "company-wide bug stomp." They claim "exploratory testing tries to make sure the software doesn't accidentally do things it isn't supposed to do." If I simply google the internet, then the definitions of exploratory testing are quite different. As a final example, in chapter 22 on retrospectives, the authors do not at all refer to any of the retrospectives techniques described in other places (like the popular "Agile Retrospectives" book).The second reason why I wouldn't recommend the book is because I personally disagree with a lot of the recommendations done by the authors. These are too numerous to all mention, but to just pick a couple. Having a separate team define the 'process' for the Agile team seems odd to me. Doing a pilot project with only 2 iterations is somewhat short from my perspective. Especially as things like velocity aren't that useful on projects like that. The iteration 0 and the adapt week are things I wouldn't recommend and certainly not on a 2 iteration project. The description of retrospectives are certainly not how I would do them myself. The focus on engineering practices is somewhat shallow. And the list goes on and on. Fair enough, this might be just my opinion against their opinion, but because of this, I would personally not recommend this book.I decided to go for two stars. Three stars would mean the book does what it is suppose to do, and I don't think it does. One star would be too low as the idea of the book is good. Also, I like the way the authors have build up their case study and the way they describe a project based on their own experiences. The book is not all bad or useless, just not the book I would recommend. Therefore decided to go for two stars.

I will be honest and say that this is the first "work related" book that actually captured my attention to the point that I would sit and read whilst my loved ones were enjoying their sweet slumber (and obviously blog about it).This book is highly informative; being neither condescending, nor pushy and over-bearing on the way that agile adoption should be approached. Without intending to be a partisan; I would even go as far as saying that it is more than recommended reading, it should be required reading for anyone thinking of migrating to agile; recent adoptees of agile or anyone in between.From the first chapters that guide us through the principles (I loved the annotated 12 principles of the Manifesto for Agile Software Development - section 1.1.2) and the descriptive text on the "paradigm shift from a plan-driven mentality"; to the readiness assessments and the importance of obtaining executive support; through to the population of the product backlog and on

to the first iterations of an example company adopting Agility in their software development process.There is no substitute for actually reading the book, which is a must; however various chapters and sections drew my attention more than others.Section 1.1.2 - The agile principles:Although the manifesto has been analyzed and described on various blogs and books; the descriptive text for each principle was really handy when evangelizing on agile and is a good primer for anyone new to the subject (and possibly even for those of us who have been practicing agile).Section 1.2 - A paradigm shift from a plan-driven mentality:This is a must read for those of us who have come from years of waterfall and attempts at changes to "traditional" methodologies or processes. The section highlights the change in mentality that is needed to move from more traditional ways of software development to a leaner, more agile way.Section 2.4 - What does it look like when a team "becomes agile":With comparative plan related diagrams of how the change is made from waterfall to agile and the breakdown of how that transition was made; is comforting for those that have made the change as well as those that are thinking about/planning the change.Section 3.2 - The different flavors of agile:This section has a really good breakdown of strengths and weaknesses of the two foremost agile methods; Scrum and XP.Section 3.3 - Create your own flavor to become agile within your constraints:When I first started in agile there seemed to be a common mantra that permeated throughout the meetings and blogs that effectively stated if you weren't doing it exactly by the book, you weren't being agile. It is completely preposterous.How can the same agile development methods used by a web advertising company be cookie-cut for a software company writing client server applications in a strong regulatory environment? The answer is - it can't. There will need to be some adaptation. After all isn't that one of the principle concepts of agile; what is the point of having Sprint Retrospectives to adjust and tweak a process in a constant attempt to improve? This section is a must read for those that are constantly being belittled or set-upon by "those in the know".Chapter 4 - The fitness test: all about readiness assessments:Although I have not had the opportunity of actually performing this exercise; having read through the chapter it is very informative and goes in to great detail on how a company can be measured for it's fitness to adopt agile development.Section 6.3.1 - Tough questions:I liked this section because of the example of typical questions asked by the team that will be adopting agile development and the possible answers to give about the migration process.Section 7.1.1 - Attributes of a good coach:This section contains a very good breakdown of the attributes that should be sought when selecting an agile coach. The following sections go into further detail about training and coaching; as well as details on the interaction with managers and stakeholders.Section 7.3 - Creating a team with an agile mindset:This section explains what ingredients are needed to create a

sound agile team; from "Culture and roles" through to "Characteristics that influence individual performance".Chapter 8 - Injecting agility into your current process:Just the title alone gives the idea of the maturity of mind that the authors have when confronting a company wanting to adopt agile development methods. From documenting the existing process (not everyone has it written in stone and perhaps not all aspects of the company are actually doing what it says in the process hand-book); to deciding what to keep and what to change.Chapters 9, 10 and 11Discuss the selection of a pilot project and the team that will be on the pilot project. This is very informative for those companies that have made up their mind to move to agile and are looking around for the best project to guinea pig.Section 12.3 - Feature cards compared to...:As a Scrum practitioner it was interesting to read the comparisons between feature cards, user stories, use cases and functional specifications.Section 12.5 - Hard-copy vs. electronic cards:I found this section particularly useful as it highlighted the benefits of both, using SharePoint as an example of the electronic format. Although the idea is to keep it as simple as possible I am an advocate of using technology where possible - more than anything to cut down on the waste generated by so much paper; but also for the fact that you can view the information at any time (this is effective when trying to explain PBI or SBI to persons not located in the same area).Chapter 13 - Prioritizing the backlog:This is a subject that I find is the least covered in discussions with peers (both at work and outside) and it is really helpful to see the suggestions posed throughout the chapter. Peppered with examples of a product backlog and how to handle various items; this chapter is a good starting point for broadening the discussion.Chapter 14 - Estimating at the right level with the right people:I still remember being asked to estimate for a particular requirement when I still felt that I did not have enough information to do a sound enough job of it. How different it is when you are doing it as a team and everyone can voice their thoughts on the matter. This chapter is particularly useful for those still in "traditional" methods or recent adopters of agile methods. Of particular interest is section 14.1 - Contrasting traditional and agile estimation techniques and section 14.2 - The importance of whole-team estimation; both of which ease the reader into the subject that is quite explanatory with regards to the estimation process in agile.Section 16.3 - Identifying and estimating tasks:This section has a nice "call-out" titled "Task assignments aren't permanent" that describes how the early stages, specialist people will be suited to certain tasks and how with the passage of time and acquired experience in an agile environment team members will become more capable of taking on varied tasks that perhaps were out of their realm in the beginning.The latter chapters of the book, starting with chapter 20, describe the process of adapting to change and learning as we go. Section 20.1 - Common reasons for adapting (i.e. make those tweaks mentioned earlier); describes common

issues that crop up and some ways that teams adapt to the situation described.Chapter 22 is particularly interesting with a detailed explanation on how to prepare and manage the retrospective.Chapter 23 is a must read for companies that have successfully started a pilot project and are now looking at how to push across the divide and bring about change at an enterprise level.

I was fortunate to get access to an early version of this book, and as I write this I'm actually going through the book jotting down some key points as our development and project management teams will be meeting to talk about how we can adopt a more agile approach to software development.We've borrowed a few agile concepts from back in the day, but we're looking to take our game to the next level. And the problem with a lot of the old school agile material is that it is very theoretical from the perspective of agile purists.Well I don't know about you guys... but it's not that simple. Change takes time, and there are business realities to factor in. You can't take a 5000 ton cargo ship moving at maximum speed to suddenly turn because you want it to.Businesses have a momentum to them, and to introduce change requires shifting that momentum. This is where the book Becoming Agile is a critical asset to folks like me (technical management), because it addresses these realities head on and provides a practical and viable approach from a business process perspective on how to get your teams developing in an agile manner.I really can't praise this book enough, and I'm aiming to get everyone in our project management and software development teams a copy of this book, and make it our guide.

Agile: Agile Project Management CherryTree Style Guide(Scrum,Agile Scrum,agile methodology,Agile development,agile coaching,agile leader,agile methods,scrum master certification,agile introduction) Agile Product Management: (Box Set) Agile Estimating & Planning Your Sprint with Scrum and Release Planning 21 Steps (agile project management, agile software ... agile scrum, agile estimating and planning) Agile Project Management: Box Set - Agile Project Management QuickStart Guide & Agile Project Management Mastery (Agile Project Management, Agile Software Development, Agile Development, Scrum) Agile Estimating & Planning Your Sprint with Scrum (agile project management, agile software development, agile development, agile scrum, agile estimating and planning) Agile Project Management: QuickStart Guide - The Simplified Beginners Guide To Agile Project Management (Agile Project Management, Agile Software Development, Agile Development, Scrum) Agile Project Management: & Scrum Box Set - Agile Project Management QuickStart Guide & Scrum QuickStart Guide (Agile Project Management, Agile Software ... Scrum, Scrum Agile, Scrum Master) Agile Product Management: (Box Set) : Scrum: A

Cleverly Concise Agile Guide and Agile: The Complete Overview of Agile Principles and Practices (scrum, ... development, agile software development) Agile Project Management: Mastery - An Advanced Guide To Agile Project Management (Agile Project Management, Agile Software Development, Agile Development, Scrum) Becoming Agile: ...in an imperfect world Agile Project Management: For Beginners - A Brief Introduction to Learning the Basics of Agile Project Management (Agile Project Management, Agile Software Development, Scrum) Agile Product Management: (Box Set): Agile Estimating & Planning Your Sprint with Scrum & User Stories 21 Tips (scrum, scrum master, agile development, agile software development) Agile Project Management: The Agile PMO: Leading the Effective, Value Driven and Agile Project Management Office (Agile Business Leadership Book 1) Agile Project Management: Agile Revolution, Beyond Software Limits: A Practical Guide to Implementing Agile Outside Software Development (Agile Business Leadership, Book 4) Agile Project Management: An Inclusive Walkthrough of Agile Project Management (Agile Project Management, Agile Software Developement, Scrum, Project Management) Agile : Agile Project Management, A QuickStart Beginners 's Guide To Mastering Agile Project Management ! Agile Product Management (Box Set): User Stories & Product Backlog 21 Tips (scrum, scrum master, agile development, agile software development) Agile Product Management: User Stories: How to capture, and manage requirements for Agile Product Management and Business Analysis with Scrum (scrum, ... development, agile software development) Agile Product Management: Product Vision:: 21 Steps to setting excellent goals for your product (scrum, scrum master, agile development, agile software development) Agile Product Management: User Stories & Product Backlog 21 Tips (scrum, scrum master, agile development, agile software development) Agile Product Management: Release Planning: 21 Steps to plan your product releases from a product vision with Scrum (scrum, scrum master, agile development, agile software development)