Agile Management for Software Engineering: Applying the Theory of Constraints for Business Results

By David J. Anderson

Agile Management for Software Engineering: Applying the Theory of Constraints for Business Results By David J. Anderson

This text explains how agile software development methods produce better business results. It aims to help managers combat the biggest business complaints about software (for example, late, doesn't deliver as promised, over-budget and so forth).

 Download Agile Management for Software Engineering: Applyin ...pdf

 Read Online Agile Management for Software Engineering: Apply ...pdf

Agile Management for Software Engineering: Applying the Theory of Constraints for Business Results

By David J. Anderson

Agile Management for Software Engineering: Applying the Theory of Constraints for Business Results
By David J. Anderson

This text explains how agile software development methods produce better business results. It aims to help managers combat the biggest business complaints about software (for example, late, doesn't deliver as promised, over-budget and so forth).

Agile Management for Software Engineering: Applying the Theory of Constraints for Business Results By David J. Anderson Bibliography

- Sales Rank: #1210676 in Books
- Published on: 2003-09-27
- Released on: 2003-09-17
- Original language: English
- Number of items: 1
- Dimensions: 9.00" h x .80" w x 6.90" l, 1.26 pounds
- Binding: Paperback
- 336 pages

Editorial Review

From the Back Cover

"This book does a good job of describing the methods employed at Sprintpcs.com ... over 250 people practicing Feature Driven Development and reporting their progress to me at the monthly operations review."
--Scott B. Relf, Chief Marketing Officer, Sprint PCS
"A tremendous contribution to the literature in the field. This should be required reading for all development teams going forward."
--John F. Yuzdepski, VP & GM, Openwave Systems
A breakthrough approach to managing agile software development, Agile methods might just be the alternative to outsourcing. However, agile development must scale in scope and discipline to be acceptable in the boardrooms of the Fortune 1000. In *Agile Management for Software Engineering,* David J. Anderson shows managers how to apply management science to gain the full business benefits of agility through application of the focused approach taught by Eli Goldratt in his Theory of Constraints.
Whether you're using XP, Scrum, FDD, or another agile approach, you'll learn how to develop management discipline for all phases of the engineering process, implement realistic financial and production metrics, and focus on building software that delivers maximum customer value and outstanding business results.Coverage includes:

- Making the business case for agile methods: practical tools and disciplines
- How to choose an agile method for your next project
- Breakthrough application of Critical Chain Project Management and constraint-driven control of the flow of value
- Defines the four new roles for the agile manager in software projects-- and competitive IT organizations

Whether you're a development manager, project manager, team leader, or senior IT executive, this book will help you achieve all four of your most urgent challenges: lower cost, faster delivery, improved quality, and focused alignment with the business.

About the Author

DAVID J. ANDERSON has been in the software business for more than 20 years, with experience as a developer and manager in start-up environments and in three of the world's largest companies. He was a member of the team that created Feature Driven Development. David is currently Director of Emerging Technology with 4thpass Inc., a Motorola subsidiary based in Seattle, WA.

Excerpt. © Reprinted by permission. All rights reserved.

Foreword

It is so good to finally have a book targeted at the software industry that challenges some of the basic business assumptions behind software engineering, and particularly those behind managing software organizations. At the time these words are written, the software business is facing huge difficulties worldwide. I hope that these difficulties also generate a willingness to look afresh at the business and to have

the courage to contemplate changes. Other industries, particularly in manufacturing, went through such conceptual changes in their business processes during the 1980s and the 1990s. It is certainly not easy, but as I've personally experienced, it is highly desirable.

In 1985 I managed my own software company in Israel and was quite proud with my new package for certified public accountants. But, even though my package competed very nicely in the market, I noticed an on-going business problem: More and more development was needed to keep the package alive. In such a case, how do I justify the on-going investment? Eventually, I was not sure that a small software company, focused on a specific market niche, could be a good business--even when the product itself was enthusiastically accepted by the market. I felt that even though I already had my MBA, I needed a better perspective to understand the business.

Then I met Dr. Eli Goldratt.

I had heard a lot about Dr. Goldratt's international software company, Creative Output, Inc., which was seen as much more than just an excellent and innovative software company. It challenged some of the most sacred norms of business, such as the concept of product cost. I could not understand how anyone could challenge the basic concept that a unit of a product has a certain cost associated with it. I was intrigued enough to be open to an offer: Join Creative Output in order to develop a video game for managers that would deliver some new managerial ideas. At the time, computerized games were focused on fast fingers and perfect coordination. They were certainly not something of interest to adults. How could a computer game be readily accepted by grown-up managers and deliver new managerial ideas?

This was the start of a mental voyage into a new management philosophy that does not lose its grip on reality. I turned myself into a management consultant with a focus on improving whatever is the particular goal of the organization. Software became an important supporting tool, but not the focus of the change efforts.

The relevance of the Theory of Constraints (TOC) to the software industry is twofold:

1. Vastly improving the flow of new products to the market.
2. Determining the real value of a proposed project, or even just a feature, to the final user. The underlying assumption is that if we know the real value to the user, it is possible to develop the right marketing and sales approach to materialize the value to the user and to the software organization.

David Anderson focuses mainly on the first aspect in this book, which includes looking at the business case and ensuring the ability to make it happen. Software organizations can definitely be improved with the help of the new generic managerial insights that have already changed traditional western industries. David does a great job in bringing together the generic managerial ideas and rationale and combining them with software-focused approaches to come up with a coherent approach on how to improve the business.

Read this book carefully with the following objective: Learn how to make more with less. Don't accept every claim David raises just because he says it is so. If you truly want to make more with less, you need to be able to internalize the claim. All I ask is you give it a chance. Dedicate time in order to rethink your environment, and then see for yourself what to do. Overcoming inertia is the biggest challenge of any really good manager in any organization. Of course, rushing to implement new fads can be even worse. Keeping an open mind and confronting new ideas that invalidate basic assumptions are what I suggest you strive for. This book is for you to struggle with. It is not trivial, and it is not a fad. If you like what you do now, it should be your responsibility to check out new ideas that might yield huge improvements.

Here are some brief insights regarding the assessment of the value to a potential customer of a new feature,

particularly to a new software package.

A new Feature can bring value to the user only if it eliminates, or vastly reduces, an existing limitation. The amount of the value depends on the limitation removed--not on the sophistication of the feature itself. Let us take a simple example. At a certain time in the history of word processors, somebody had an idea: Why not add a spell checker to the package?

What is the value of the spell check we now have as a routine feature? What limitation does it eliminate or reduce? For people with a very good knowledge of the language, spelling mistakes are caused by writing too fast. So, without a spell checker, those people need to read carefully what they just wrote. People who are not in full command of the language (for example, me, as an Israeli) need to look at the dictionary very often, which is quite time consuming.

This need leads us to recognize two additional insights.

People developed some rules to help them overcome the limitation. People who used word processors had to go over whatever they just wrote before sending the document to others. People without good command of the language needed to be supported by a dictionary.

Once the limitation is vastly reduced, people should replace the old rules with new ones that take full advantage of the removal of the limitation. If this does not happen, then there is no added value to the Feature.

Now we can see whether adding a spell checker to an existing word processor brings value. Suppose you have perfect command in the language, would you now refrain from carefully reading your recent document before sending it away? Spelling mistakes are hardly the main reason to go over any document that I want other people to read. So, for people with perfect knowledge, the spell checker offers no real value. But, for me as a person in good command of Hebrew, but not good enough in English, spelling mistakes in English are a nuisance. But, could I really avoid them just by the use of a spell checker? As long as the spell checker does not suggest how to write the word correctly--the limitation is only marginally reduced and thus not much value is produced. This means that if we want to generate significant value for the specific user group that has not mastered the language, we need to add good suggestions of what should be written instead.

In this simplified example, we already see the need to check the behavior rules both before the limitation is eliminated and after. Is it always clear what the new behavior rules should be? Assuming the user is well aware of what the new rules should be is a very common trap for too many software features.

Suppose that a new Feature is added to a sales-graph display module in which the trends shown by the graph are analyzed for statistical significance. The limitation is lack of knowledge on whether market demand is really up or down or just part of the normal statistical fluctuations. The current behavior of the management is: If sales are up, the sales agents are complimented and get appropriate bonus; if sales are down, there are no bonuses and some hard talk from management.

What should the new management rules be once management knows whether the rise in sales is significant? I'm afraid that in the vast majority of the cases the behavior will be exactly the same. Hence, the newly added Feature will not add value to the customer, even though some customers might ask for it and even exert a lot of pressure to have the Feature developed. Eventually, the value to the software company of developing the Feature will be negative.

Of course, for a good managerial consultant assisting in the formation of better decision processes, a specific Feature can bring immense value both to the consultant and the client. In this case, a strategic partnership

between the consultant and the software company can be a win-win for all, including the client.

Improving the flow of the Features that truly bring value to the customer and also have a good chance of generating revenues for the software organization is what this unique book is all about. The Agile Manifesto principle of "Our highest priority is to satisfy the customer through early and continuous delivery of valuable software" is fully in line with the Theory of Constraint's objectives. To be more precise, TOC strives to generate more of the organization's goal. But, in order to do so, the organization has to generate more value to its customers. The means of early and continuous delivery of software that truly generates value should assist both the software organization and its clients in achieving more of their respective goals.

Please bear in mind that improvement has only one criterion: Achieving more of the goal. The path to truly improving the performance of your software organization may well start with this book.

--Eli Schragenheim

Users Review

From reader reviews:

Carol Castaneda:

Have you spare time for a day? What do you do when you have considerably more or little spare time? Sure, you can choose the suitable activity for spend your time. Any person spent their particular spare time to take a walk, shopping, or went to the Mall. How about open or perhaps read a book called Agile Management for Software Engineering: Applying the Theory of Constraints for Business Results? Maybe it is being best activity for you. You understand beside you can spend your time along with your favorite's book, you can more intelligent than before. Do you agree with it is opinion or you have additional opinion?

Mark Bottoms:

Information is provisions for anyone to get better life, information these days can get by anyone with everywhere. The information can be a expertise or any news even a concern. What people must be consider if those information which is inside the former life are hard to be find than now is taking seriously which one is acceptable to believe or which one often the resource are convinced. If you find the unstable resource then you understand it as your main information you will have huge disadvantage for you. All those possibilities will not happen throughout you if you take Agile Management for Software Engineering: Applying the Theory of Constraints for Business Results as the daily resource information.

Helen Arnold:

Do you really one of the book lovers? If yes, do you ever feeling doubt while you are in the book store? Make an effort to pick one book that you find out the inside because don't evaluate book by its cover may doesn't work the following is difficult job because you are frightened that the inside maybe not while fantastic as in the outside search likes. Maybe you answer can be Agile Management for Software Engineering: Applying the Theory of Constraints for Business Results why because the excellent cover that make you consider with regards to the content will not disappoint anyone. The inside or content is usually fantastic as the outside or maybe cover. Your reading sixth sense will directly assist you to pick up this book.

Robert Wallace:

That e-book can make you to feel relax. This specific book Agile Management for Software Engineering: Applying the Theory of Constraints for Business Results was vibrant and of course has pictures on the website. As we know that book Agile Management for Software Engineering: Applying the Theory of Constraints for Business Results has many kinds or style. Start from kids until adolescents. For example Naruto or Private eye Conan you can read and feel that you are the character on there. Therefore not at all of book are usually make you bored, any it makes you feel happy, fun and loosen up. Try to choose the best book to suit your needs and try to like reading that will.

Download and Read Online Agile Management for Software Engineering: Applying the Theory of Constraints for Business Results By David J. Anderson #P9HRZ4G52BT

Read Agile Management for Software Engineering: Applying the Theory of Constraints for Business Results By David J. Anderson for online ebook

Agile Management for Software Engineering: Applying the Theory of Constraints for Business Results By David J. Anderson Free PDF d0wnl0ad, audio books, books to read, good books to read, cheap books, good books, online books, books online, book reviews epub, read books online, books to read online, online library, greatbooks to read, PDF best books to read, top books to read Agile Management for Software Engineering: Applying the Theory of Constraints for Business Results By David J. Anderson books to read online.

Online Agile Management for Software Engineering: Applying the Theory of Constraints for Business Results By David J. Anderson ebook PDF download

Agile Management for Software Engineering: Applying the Theory of Constraints for Business Results By David J. Anderson Doc

Agile Management for Software Engineering: Applying the Theory of Constraints for Business Results By David J. Anderson Mobipocket

Agile Management for Software Engineering: Applying the Theory of Constraints for Business Results By David J. Anderson EPub