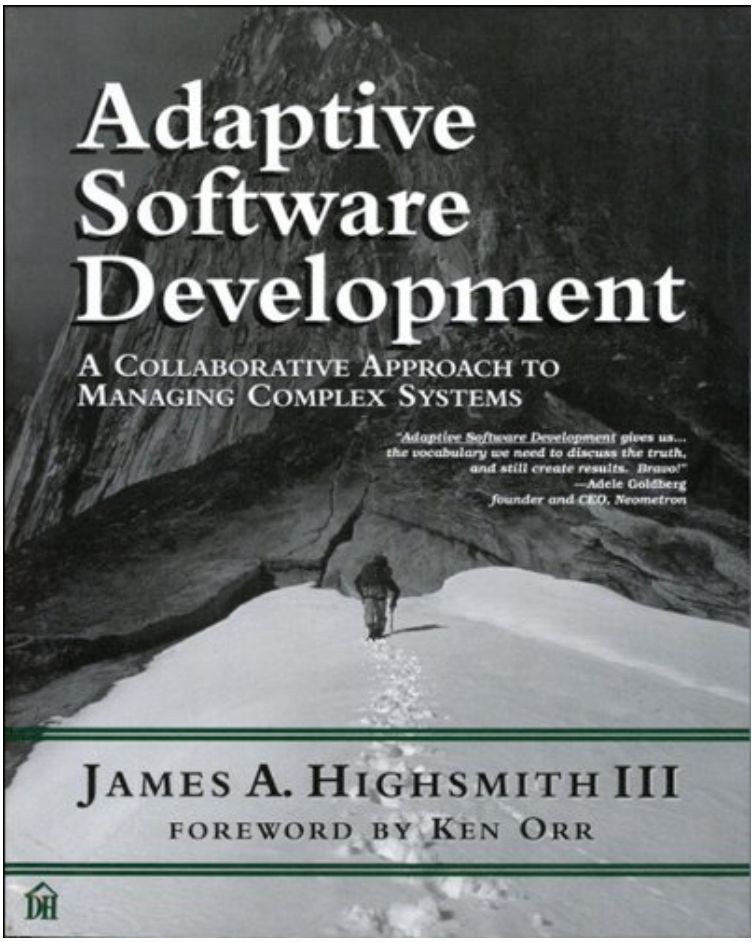
Adaptive Software Development: A Collaborative Approach to Managing Complex Systems (Dorset House eBooks)

Von James Highsmith III
*DOC | *audiobook | ebooks | Download PDF | ePub*



Produktinformation -Verkaufsrang: #700096 in eBooksVerffentlicht am: 2013-07-15Erscheinungsdatum: 2013-07-15File Name: B00DY3KQSQ | File size: 56.Mb

Von James Highsmith III : Adaptive Software Development: A Collaborative Approach to Managing Complex Systems (Dorset House eBooks) before purchasing it in order to gage whether or not it would be worth my time, and all praised Adaptive Software Development: A Collaborative Approach to Managing Complex Systems (Dorset House eBooks):

KundenrezensionenHilfreichste Kundenrezensionen4 von 4 Kunden fanden die folgende Rezension hilfreich. Mountain climbing as an analogy for software developmentVon Charles AshbacherThere are many areas of human endeavor that can be used as an analogy for software development. In this book, the author uses mountain climbing to illustrate his points about teamwork, planning and adaptation to rapidly changing conditions. The points are well-taken, although he does stretch it a bit. If the team doesn't function well or a judgment error is committed while climbing, there is the real and immediate threat of injury or death. Similar problems in software development lead to

much gentler consequences that are sometimes years in the future. One does not easily change teams in the middle of a climb and developers often have several golden ropes to clutch if it is necessary to leave. Nevertheless, the comparison is largely a good one. The most significant point is about how software development must be a process of aggressive, rapid adaptation to changing conditions. Among all the things that we do, software construction changes faster than anything else. The solution is to perform the delicate act of balancing on the head of a pin. On one side, there is the necessity of setting down standards of rigor that will keep the process within acceptable boundaries. However, the addition of too much rigor and the mortis sets in, making it too difficult to change the product when the inevitable modifications are needed. Many such strategies for how to maintain this minuscule middle are set forward. There are many points of sound advice in this book, several of which lead to the following simple adage. "Rules can be barriers to hide behind or guidelines for the wise to consider and break when the circumstances justify it." Effectively executing the latter is the not so secret plan for success in the current IT world of dynamic competition. I respectfully disagree with the author on one point. He argues that the day of the lone "coding cowboy", where one programmer builds a killer app, is over. While the lone programmer may be a thing of the past, the small team is not, and some small teams of two or three can work wonders. The second point of my disagreement are due to the increasing use of components and rapid development tools. Given the library of tools that can be used, it is now possible for one person to put several distinct items together in a unique way and build a complete system. Furthermore, it may be possible for a lone programmer to build the next little "big thing" component that could revolutionize how we do some things. Despite my occasional disagreements with the points made, reading this book made me think a little harder about some aspects of the software development process. The authors' metaphors of biological adaptation and mountain climbing have many equivalencies in software development that should be seriously considered.

Kurzbeschreibung This is the digital version of the printed book (Copyright 2000). Winner of the 2001 Software Development Jolt Product Excellence Award This innovative textoffers a practical, realistic approach to managing high-speed, high-change software development projects. Consultant James A. Highsmith shows readers how to increase collaboration and adapt to uncertainty. Many organizations start high-speed, high-change projects without knowing how to do themand even worse, without knowing they dont know. Successful completion of these projects is often at the expense of the project team. Adaptive Software Development emphasizes an adaptive, collaborative approach to software development. The concepts allow developers to scale-up rapid application development and extreme programming approaches for use on larger, more complex projects. The four goals of the book are to support an adaptive culture or mindset, in which change and uncertainty are assumed to be the natural statenot a false expectation of order