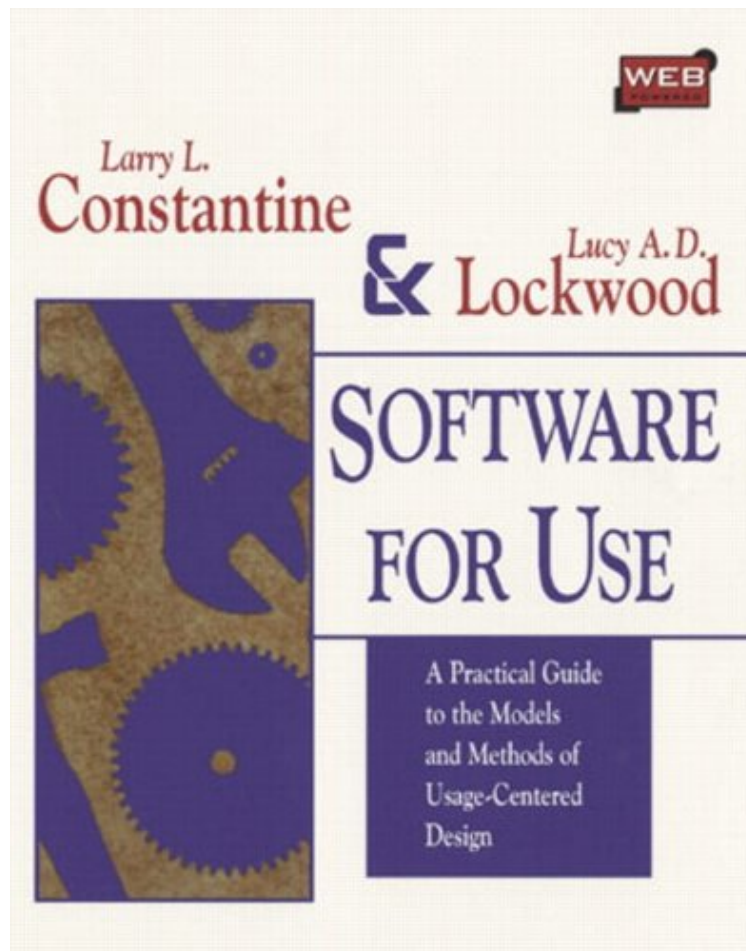


(Download free pdf) Software for Use: A Practical Guide to the Models and Methods of Usage-Centered Design

Software for Use: A Practical Guide to the Models and Methods of Usage-Centered Design

Von Larry L. Constantine, Lucy A.D. Lockwood
audiobook | *ebooks | Download PDF | ePub | DOC



[Download](#)

[Read Online](#)

Produktinformation -Verkaufsrank: #424064 in eBooksVerffentlicht am: 1999-04-07Erscheinungsdatum: 1999-04-07File Name: B001EWOG7Q | File size: 64.Mb

Von Larry L. Constantine, Lucy A.D. Lockwood : **Software for Use: A Practical Guide to the Models and Methods of Usage-Centered Design** before purchasing it in order to gage whether or not it would be worth my time, and all praised Software for Use: A Practical Guide to the Models and Methods of Usage-Centered Design:

KundenrezensionenHilfreichste Kundenrezensionen4 von 4 Kunden fanden die folgende Rezension hilfreich. A must-read for anyone involved with user interfacesVon Ein KundeI very much enjoyed Software for Use. When I got the book, I read it from cover to cover. Now, six months later, I still turn to it regularly as I develop use cases for our application - especially when I'm working on the user interface for each use case.You might expect that anything from Larry Constantine would be terrific and again he - this time with equally adept co-author Lucy Lockwood - hasn't let us down. SfU (as it seems to be called in the chat groups) addresses one of the most under-addressed issues in our

industry: Why is so much potentially useful software in fact useless because of its appalling user interfaces? There are lots of valuable topics in this book. But probably the most valuable thing that I got from the book was a methodical approach to developing user interfaces, through the user role maps, task models, essential use cases, use case narrative, tools and materials and ... well, perhaps you should read the book! If you're in the mood for some edutainment (i.e. cheap laughs while learning something handy), read the section on Web wisdom. There you'll find some wonderfully silly interface designs to avoid on your next e-commerce project. Bottom line: I'll bet this book's on the way to becoming a classic. It's a "must read" for every software developer involved in any way with software that's meant to be used. And if you're actually charged with developing user interfaces, I'd say it's a "must own".

1 von 1 Kunden fanden die folgende Rezension hilfreich. I bought a copy for each person on my team!

Von Ein Kunde I've been an object-oriented bigot since the late eighties when I first stumbled on Betrand Meyer's "Old Testament" entitled: "Object Oriented Software Construction". Since then I have been on a continual methodology quest, picking and choosing what works from each methodology. When I finally came across Larry's book it felt like coming home. He has done an amazing job of amalgamating all the things that in my experience work. How many times have I argued with engineers about user interface design! How many times have they told me that they know better, and oh by the way, look at this cool feature we added (hit shift F6 and it does this...) This is the best book I have found so far when it comes to usage-centered design. I was lucky enough to be at the start of a new project and bought a copy of the book for every team member! This book has become our baseline and the quality of our software will reflect it. If you want to build better user interfaces you should buy this book!

0 von 0 Kunden fanden die folgende Rezension hilfreich. Software für den Menschen sollte klar sein, leider ist es nicht, und dieses Buch hilft, gute Software wahrscheinlicher zu machen.

Von Charles Ashbacher Die besagt, dass Software ist es, für den Menschen zu bedienen geschrieben werden ist ein Satz, eine sarkastische Reaktion in Frage die Intelligenz der einen uert sie provozieren sollte. Wre dies immer der Fall ist, dann gibt es wahrscheinlich ein Comic ähnlich dem uerst beliebten Dilbert diesem Thema gewidmet sein. Leider Zeitdruck, Komplexität, Unwissenheit und niedrige Erwartungen der Nutzer zu kombinieren, um es nur allzu blich, dass der Ausdruck "user friendly" könnte genauso gut ein Mantra in Wal Lied zu machen. Einige Projekte werden ohne jegliche Eingabe von denen als das Zielkundenbasis entwickelt, in anderen wird der Eingang nur erbeten, um ignoriert oder verharmlost werden. Die Autoren dieses Buches argumentieren, für die Einfügung von was sie als "Nutzung-Centered Design" in den intensiven, oft chaotischen Prozess der Erstellung von Software.

Kapitel 11, "Helfen Sie mir, wenn Sie können: Entwerfen Hilfe und nützliche Meldungen", machte den größten Eindruck. In der Geschichte der Software-Entwicklung, die Schaffung von wirklich hilfreiche Dokumentation ist eine Kunst beklagens MIA (durcheinander, unverständlich, und abstruse). Auch die intuitive Schnittstellen mit gelegentlich verlassen Sie verwirrt, oder Sie werden auf dem falschen Element klicken, oder drücken Sie eine unangemessene Taste. Die Entwicklung der wesentlichen Anwendungsfälle für Hilfe bringt das Thema in acht Aussage Präfixe, die alle Eltern eines jungen Kindes steht mehrmals am Tag. Einige der besten Ratschläge für Entwickler von Online-Hilfe kommt von den Grundprinzipien des Zeitungsjournalismus. Komm zum Punkt. Die ganze Geschichte erzählen in der berschrift. Die ganze Geschichte erzählen im ersten Absatz. Oder die noch stumpfer Beratung, "Machen Sie Ihre Hilfe Hilfe." Der bedeutendste Fortschritt beim Entwurf von Benutzerschnittstellen hat die Schaffung des Symbols. Ein Abschnitt "" eye-con "Design" bringt die Dinge auf den Punkt - es ist das Auge, was zählt. Mit so wenig Fläche zur Verfügung, die Schaffung von Qualitäts Icons ist eine echte Kunst. Allerdings ist es etwas, das gelernt werden kann. Dies ist ein Abschnitt, die erforderlich sein sollten, bevor Sie die Lesezeichenprogramm zu öffnen.

Kein Buch, das den Benutzer fordert kann komplett ohne umfassende Behandlung des World Wide Web sein. Zwar gibt es keinen Zweifel, dass das Web hat die Welt verändert, so richtig darauf hingewiesen, Programmierung für das Internet unterscheidet sich nicht allzu sehr von der "traditionellen" Programmierung. Der Ansatz der Autoren ist in der berschrift zusammenfassen: "Das Chaos und der Komplexität des Web stellt eine Prämie auf solche altmodischen Tugenden machte deutlich, um Benutzer, wo sie sind und wie sie dorthin gekommen." Ich würde die zusätzliche Zeile hinzufügen ", und klar zeigen ihnen, was sie tun sollten." Diese Philosophie wird von Erfahrung, wo die Nutzer in der Regel nicht länger warten, als die Anzahl der Sekunden, können sie an den Fingern abzählen gesichert. Benutzer auch vermeiden Websites, die schwierig zu navigieren oder als unklar, als ob sie in monochromen waren sind. Wenn es eine Schwäche in diesem Buch ist es der Mangel an klaren Beispielen. Wie der Autor Highlights an mehreren Stellen, in ihrem Job gut zu funktionieren viele Menschen, ohne zu denken viel darüber. , Wenn sie aufgefordert werden, zu kodifizieren, was sie tun und wie sie es tun, ergeben sich jedoch Probleme. Nur klare Beispiele können zeigt den wahren Weg zum Verständnis. Dies ist besonders erzählt im Abschnitt über Web-Design. Zwar gibt es eine Darstellung eines schlecht konzipiert Onlineformular gibt es kein Beispiel für eine gute. Ein Link auf eine Website zeigen, eine gut gestaltete Form wäre eine enorme Verbesserung. Es gibt eine Webseite auf der Rückseite aufgeführt, aber es ist der Ort der Unternehmen, in dem die Autoren zu arbeiten und jedes Material direkt zu diesem Buch ist nicht ohne weiteres ersichtlich. Es gibt so viel wertvolles Material in diesem Buch, die die einzig mögliche ist, um nur einige der bemerkenswertesten Elemente zu markieren. Falls und wenn ich jemals wieder in der chaotischen Welt der Software-Entwicklung, Einkauf Kopien für jedes Mitglied des Teams wird in den Haushalt geschrieben werden, zusammen mit der notwendigen Zeit zum Lesen und diskutieren Sie - wahrscheinlich die höchste Rendite, die das Projekt haben

KurzbeschreibungIn the quest for quality, software developers have long focused on improving the internal architecture of their products. Larry L. Constantine--who originally created structured design to effect such improvement--now joins with well-known consultant Lucy A. D. Lockwood to turn the focus of software development to the external architecture. In this book, they present the models and methods of a revolutionary approach to software that will help programmers deliver more usable software--software that will enable users to accomplish their tasks with greater ease and efficiency. Recognizing usability as the key to successful software, Constantine and Lockwood provide concrete tools and techniques that programmers can employ to meet that end. Much more than just another set of rules for good user-interface design, this book guides readers through a systematic software development process. This process, called usage-centered design, weaves together two major threads in software development methods: use cases (also used with UML) and essential modeling. With numerous examples and case studies of both conventional and specialized software applications, the authors illustrate what has been shown in practice to work and what has proved to be of greatest practical value. **Highlights** Presents a streamlined process for developing highly usable software Describes practical methods and models successfully implemented in industry Complements modern development practices, including the Unified Process and other object-oriented software engineering approaches .deFor anyone who designs applications or Web pages professionally, **Software for Use** provides an appealingly written guide to user interface design. This book delivers many valuable insights on improving interfaces for both desktop applications and Web sites. A software design process is first presented that's centered on usability (with terms like "user roles," "use cases," and "interaction contexts"). Early sections have much to say about inadequate interfaces (using a number of Windows examples) and how to improve them. The book presents an argument for creating innovative and intuitive interfaces (often by rethinking time-honored Windows conventions). The most provocative material here is the coverage of the Internet--the book argues that many Web sites sacrifice usability for visual razzle-dazzle, and it offers ways to organize Web sites for better usability. (A section on embedded devices looks at UI issues for these systems too.) A full case study of a user interface design for a corporate address book is included. **Software for Use** makes a good case that there is room for improvement in today's user interfaces. This book is sure to be a valuable resource for anyone serious about improving the user's experience of software or Web sites. --Richard Dragan **Topics covered:** Design processes; help systems and error messages; and interface creation for novice, intermediate, and advanced users.**Kurzbeschreibung**In the quest for quality, software developers have long focused on improving the internal architecture of their products. Larry L. Constantine--who originally created structured design to effect such improvement--now joins with well-known consultant Lucy A. D. Lockwood to turn the focus of software development to the external architecture. In this book, they present the models and methods of a revolutionary approach to software that will help programmers deliver more usable software--software that will enable users to accomplish their tasks with greater ease and efficiency. Recognizing usability as the key to successful software, Constantine and Lockwood provide concrete tools and techniques that programmers can employ to meet that end. Much more than just another set of rules for good user-interface design, this book guides readers through a systematic software development process. This process, called usage-centered design, weaves together two major threads in software development methods: use cases (also used with UML) and essential modeling. With numerous examples and case studies of both conventional and specialized software applications, the authors illustrate what has been shown in practice to work and what has proved to be of greatest practical value. **Highlights** Presents a streamlined process for developing highly usable software Describes practical methods and models successfully implemented in industry Complements modern development practices, including the Unified Process and other object-oriented software engineering approaches