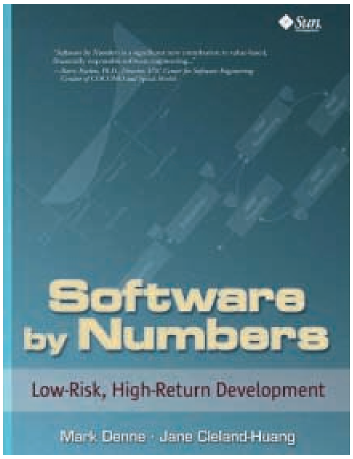
Good Reading

ARE YOU STARTING THE new year resolved to brush up on your software skills? If you've browsed the Internet lately looking for a good book on building better software, you know that the number of options can be a bit overwhelming. Before you click that "buy" button, how can you be sure you're getting something worth reading? That's where StickyMinds.com comes to the rescue with its Books Guide providing descriptions, ratings, and reviews for hundreds of software books. Here are some recent reviews written by StickyMinds members Jennifer Cannon and Mark L. Krug.



Software by Numbers: Low-Risk, High-Return Development

AUTHOR: Mark Denne/Jane Cleland-Huang

PAGES: 186

PUBLISHED: 2004

PUBLISHER: Prentice Hall

ISBN: 0131407287

Mark Denne and co-author Jane Cleland-Huang have assembled several years of experience in winning competitive contracts for systems integration and application development projects to create a new approach to the software development process, called the Incremental Funding Methodology (IFM).

Most people know that the software development process has risks and costs, and that much time and energy is spent trying to reduce those risks and costs. However, this book clearly shows the benefits of looking at the software development process as a value creation process. The authors take the latest ideas in iterative development methodologies and apply them in such a way that the greatest financial benefits are realized for the customer, in addition to the technological benefits he receives.

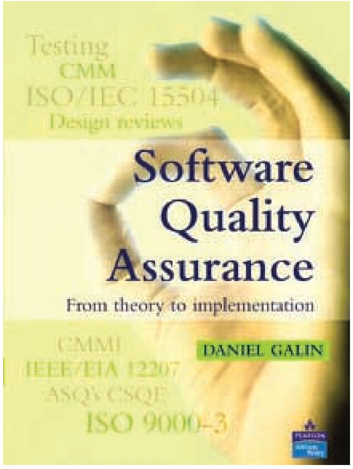
The first part of the book deals with a discussion of the financial metrics that are used to track value creation activities such as ROI, cash flow projections, payback time, net present value, breakeven time, and the internal rate of return. Next is a discussion of Minimum Marketable Features (MMF) and how they impact those financial metrics. An MMF is defined by Denne as a chunk of functionality that delivers a subset of the customer's requirements and is capable of returning value to the customer when released as an independent entity. MMFs are the building blocks of IFM, and the carefully prioritized delivery schedule of these customer-valued functionality "chunks" is what defines this methodology. The book goes into great detail about identifying MMFs, IFM sequencing strategies, and IFM heuristics. The authors include two chapters where they apply the IFM principles to the Rational Unified Process and to Agile development methodologies (XP, feature-driven development, and SCRUM), so the reader will understand more than just the academics of this methodology. There's also a case study where the IFM principles are applied to a hypothetical project, which brings all of the principles together.

In many ways iterative software development methodologies have been successful. As powerful as iterative development methodologies are, Denne and Cleland-Huang do a great job in taking the principles that make the methodologies so successful and adding a whole new dimension to the entire development process that can take software development to another level. The book has only 186 pages, but the authors clearly explain the theories and tools needed for utilizing the Incremental Funding Methodology, in addition to providing example financial calculations to support their theories. If you need further help with implementing the methodology, there's a companion website that contains additional information and downloadable tools.

The chapters in which the IFM principles are applied to the Rational Unified Process and the Agile development methodologies, as well as the case study that illustrates the principles using a hypothetical project are very helpful and provide a more complete example of how the methodology works. Instead of being bombarded with just theories and tables of numbers, you're able to see how this methodology fits hand in hand with iterative development methodologies. It's such a plus to know that in order to implement IFM, your company doesn't require a total overhaul of its current development methodology (if your company is already using an iterative development methodology, that is, if you're still using the conventional "Waterfall" approach, it may not work).

I believe this book can be a useful resource to analysts, project managers, developers, and customers.

—Review by Jennifer Cannon



Software Quality Assurance: From Theory to Implementation

AUTHOR: Daniel Galin

PAGES: 576

PUBLISHED: 2004

PUBLISHER: Addison-Wesley

ISBN: 0201709457

In *Software Quality Assurance: From Theory to Implementation,* Daniel Galin gets right to the point and does not inundate you with more information than is needed. The various charts, graphs, and examples are well placed and relevant to the subject. The materials are presented from many viewpoints and are not biased. The look and feel of the book is pleasing, but not distracting. The author is experienced, knowledgeable, and passionate about his area of expertise. In short, it is one of the best books I have read on this subject.

The book approaches the subject at a level appropriate for both the beginner and the experienced practitioner. The review questions and discussion topics help the reader fully comprehend the materials, as well as prepare for any questions that upper management might have before implementing such practices or standards.

The book also serves as an excellent quality assurance study guide and contains much of the material needed to pass either the QAI CSTE or the ASQ CSQE exams. It could also possibly serve as a study guide for the QAI CSQA exam as well. The book is roughly divided by what is covered in each test; thus making it easy to stay focused while studying.

Each chapter deals with one principle or area at a time. The chapters interconnect well, and cover the right amount of material before moving on to the next subject. The author presents many viewpoints other than his own and is not argumentative about choosing one over the other. He simply writes as an unbiased observer, while still illustrating the fashion in which he personally implements the various components. The review questions and discussion topics cover subjects the author finds to be most relevant to the practice of quality assurance.

For those trying to start or better implement quality assurance within their organization, it is an excellent resource. For those seeking certification, it is a fantastic study guide.

—Reviewed by Mark L. Krug