

Reviews

teams excessively obsess about the role of tools in successful agile delivery, the authors provide some good direction as to where tools really matter and how to select an appropriate toolset. Chapters 15 and 16 provide a view into the construction phase using a "day in the life" format, which makes it interesting to read and enables the reader to relate to their current practices. Chapter 17 provides an interesting case study covering the implementation of a point-of-sale system. This chapter effectively ties several DAD practices together.

Chapter 20 deals with governance, and while I know the very word might raise concerns for an agile purist, any medium to large organization that is working toward agile delivery will realize that effective governance, risk management, and discipline are imperative for an organization to be able to scale. Also, organizations that are interested in driving engineering and operations toward agile delivery cannot afford to minimize the role of governance. This section would also be helpful for IT organizations that use multiple reference models such as CMMI, COBIT, and ITIL and for exploring how they can stay lean and build agility into their solution delivery process.

Lastly, I would also suggest that readers visit the book's companion website www.disciplinedagiledelivery.com, which includes some good reference materials and several interesting nuggets that practitioners can readily use.

Prem Ranganath is a director and global head of testing, validation, and quality at Quintiles Inc. He is passionate about quality and stays current by presenting at conferences and by teaching an online SQA course in the MS-Computing program at Marquette University.

SYSTEMS AND SOFTWARE ENGINEERING PROCESSES

Think Like a Programmer: An Introduction to Creative Problem Solving

V. Anton Spraul. 2012.

No Starch Press
(<http://nostarch.com>).

23 pages.

ISBN: 978-1-59327-4245

CSQE Body of Knowledge
area: III.E.1 Design Methods

Reviewed by Trudy Howles
tmh@cs.rit.edu

The title of this book led me to believe it would be a general entry-level problem-solving book. However, this book is very language-specific; the problem-solving material is tightly coupled with programming examples using the C++ language.

The author presents examples of frequent problem or programming areas, many of which go beyond an introductory understanding of a language, including the OO specifics of C++ classes and objects, structs, recursion, dynamically allocated structures such as linked lists, memory management, and, of course, pointers. I would define most of these as intermediate or advanced topics, rarely fully understood by those starting with only an exposure to basic syntax and the semantics of any language.

This book might be a helpful supplemental text for developers wishing to polish up their problem-solving skills. The problems presented are thoroughly broken down. The author does provide the basics and some good tips for general problem solving, such as using what one already knows or has previously seen, and breaking down the problem into manageable pieces. The book also includes challenge problems at the end of the chapters, and a topic index.

Trudy Howles is a full professor in the Computer Science Department at the Rochester Institute of Technology in Rochester, New York. Prior to entering the teaching profession, she worked for more than 20 years as a consultant, senior software development engineer, and project leader in the banking industry, the Eastman Kodak Company and General Motors. Her current research areas include secure software and privacy preserving data mining. Howles is a CSQE, a Senior member of ASQ, and an associate editor for the *Software Quality Professional*.

Essential Scrum: A Practical Guide to the Most Popular Agile Process

Kenneth S. Rubin. 2013.

Addison-Wesley

(<http://www.pearsonhighered.com>). 452 pages.

ISBN-13: 978-0-13-704329-3

CSQE Body of Knowledge
areas: III.A Lifecycles and
process models; II Software
Quality Management; IV
Project Management; V.E.4
Software Metrics & Analysis

Reviewed by Prem Ranganath
prem.ranganath@gmail.com

Just when I thought I had read every possible interpretation and experience story based on Scrum comes a unique book from Ken Rubin that presents both experiences and guidance in a whole new light. Rubin takes a unique approach to presenting the popular delivery model, which in turn makes this book an "essential" read for managers and their teams. I have heard conflicting views from teams about Scrum—ranging from its too simple and flexible (and thereby it cannot be applied in regulated environments) to it not being adequately prescriptive and therefore not easy to adopt. Rubin has provided some elaborate discussions that will address both of these

camps. While I cannot provide an elaborate review of the book here, I wanted to highlight a few chapters and topics I found very interesting. Overall, Rubin's presentation style is very simple, easy to follow and ready to implement. The book uses a very different notation and style for the many visuals, which play a significant role in improving the overall reader experience and value proposition.

Chapter 3 is a great place for every manager to start, as this chapter builds a strong case for why Scrum matters from a risk perspective and why it presents a stronger case than big upfront planning. This chapter also provides good insight on the very important topic of WIP and its impact on controlling delivery outcomes and on people. Chapter 6, which discusses practices for managing the product backlog, provides both elaboration within the context of the controls and discipline required for managing the backlog. Anyone who is still concerned that Scrum is too flexible and lacks controls will surely reverse that assumption after reading this chapter. This chapter leads to the all-important discussion on estimation, velocity, and technical debt. While these topics by themselves may not be new, Rubin's style, which is easy to read and follow and is suitably supported by great visuals, makes a big difference.

I found some very useful guidance regarding organizations and teams in chapters 9 and 11, and I readily zeroed in on the sections that focused on globally distributed teams, which was very well presented. Chapter 9 provides a very interesting and useful discussion on the role of product managers, and chapter 11 provides very useful insight for development teams. I found Rubin's guidance on the "T-Shaped Skills" and the "Musketier Attitude" very valuable, and there are several pointers to

teams who are still figuring out their path on Scrum or to more experienced teams who are working to resolve some team issues. I would also recommend that managers and teams review the discussion on portfolio planning and key considerations for scaling as discussed in chapter 16. Chapters 14 and 15 provide a very comprehensive discussion on planning; the highlight in these chapters is the guidance in chapter 15 around multilevel planning, which also transcends very naturally into the evolving area of portfolio level planning for agile programs. For anyone with possible concerns regarding Scrum's ability to scale, especially within large global organizations, chapters 14 through 16 should help resolve those concerns. The discussion on sprint execution in chapter 20 is another area that certainly provides very interesting insight into key considerations and more than adequate guidance for effectively executing sprints. Despite all of the planning, the success of each sprint and of the overall release depends on how well teams manage execution. The ills of excessive multitasking and the popular belief that people need to always be 100 percent utilized is discussed very well in this chapter, along with a strong case for teams to be goal focused instead of task focused. The case for not being task focused further strengthens the case for not being obsessed with the need to ensure 100 percent utilization.

An overarching theme in this book is that a key measure of success is to determine if people and organizations are successful by using Scrum and agile practices. It's not about successfully executing the key tenets of Scrum end to end. A good illustration of the people perspective (there are many throughout the book) is the emotional seismograph presented in chapter 22 as part of a detailed

discussion of what requires attention during retrospective meetings. For anyone wondering why Scrum is not prescriptive and how one goes about transforming a team into being agile, chapter 23 is a good read and a very appropriate wrap-up for the book.

Rubin's approach to presenting Scrum practices enables teams to readily benchmark their current practices with a set of recommendations and to readily identify practices that can be used to address gaps. This is a great book on Scrum and I strongly recommend that emerging and experienced agilists give a serious consideration to this title as they work toward raising their capabilities in effectively using Scrum.

Prem Ranganath is a director and global head of testing, validation, and quality at Quintiles Inc. He is passionate about quality and stays current by presenting at conferences and by teaching an online SQA course in the MS-Computing program at Marquette University.

SOFTWARE VERIFICATION AND VALIDATION

IEEE Standard for System and Software Verification and Validation (IEEE Std 1012-2012)

IEEE Standards Association. 2012.

ISBN 978-0-7381-7268-2

ISBN 978-0-7381-7374-0

CSQE Body of Knowledge areas: VI. Software Verification and Validation (V&V), I.C Standards & Models

*Reviewed by Eva Freund
evafreund@ivwgroup.com*

Standards play a vital role in the arena of system and software development and are promulgated by numerous international and U.S. organizations. In addition, there