Agile Documentation: A Pattern Guide To Producing Lightweight Documents For Software Projects
He quotes a lot of others in his book and identifies that the goal (of agile documenting) is 'light-weight but sufficient'. If you're working on developing a system for documenting software development (or really any other technical project) this would make a good reference, but start at the back of the book. He uses patterns in his book as his technique to cover the topic. They are recognized problems and solutions, and he lists all of them in the back of the book in thumbnail form. The middle of the book provides the elaboration. The first part of the book is his professional history (sorry Ruping, but not that interesting), and his how-to-use-the-book section. The problem I had with the book is that I didn’t see anything new and innovating. I didn’t have an `ah-ha’ moment where I finally understood the real principles behind 'agile' documentation. I didn’t find this book ‘agile’ at all. I also did not agree with all of his solutions, most seemed cliché. But again, if you’re looking for a reference on the topic of technical documentation - this is not bad.

As a professional technical writer I was hoping to find insights on how to do my job in an Agile development environment, as my experience is in the slower-moving "waterfall" types of development. The book clearly says early on, but you couldn’t tell from the description online, that it’s really more about the design documentation that the development and test people will need to use among themselves. I wish I’d had a chance to borrow and skim for 10 minutes, I would have saved my money. Given what it is, it’s a good primer for developers to use for specs and so on. If those are well-done, the end-user docs are much easier to craft, so I’d recommend it to the intended audience.

As a tech communicator with long experience in various engineering environments, I am enthused about the concepts espoused here. We need to write "documentation" the way we like to read "documentation"; it needs to have the content needed particularly by us, in the style we need, and concise and to the point when our need begs for that. We need to look toward long term relevance and up-to-dateness - of all technical information, especially technical reference information, and it is more likely to be so if it is concise and focused to start with. We are all scanners, doing stuff or acquiring information in a hurry, doing tasks in a hurry. Another book told about how a new guy on a project was given a thick document about the project. Two days later, he had a headache; couldn’t get the picture. If he were given an agile-ly written doc, he would have gotten the picture. Ginny Redish (Letting Go of the Words) is another great one along the same thinking. ONE distractor: He wrote the book with a bit of his own fluff. Maybe to give the book a slight bit of thickness. Could have
been more to the point. An engineer I used to work with used to say "paid by the pound".

The author misleads the audience with his title, "Agile Documentation." There is nothing about Agile methodology in the book, rather a way to produce "Lightweight Documents." It is conceivable that "Lightweight Documents" could be appropriate deliverables for products using Agile development, but he does not lay that foundation.

Download to continue reading...

Agile Documentation: A Pattern Guide to Producing Lightweight Documents for Software Projects
