Software Requirements and Specifications is the latest book from Michael Jackson, one of the foremost contributors to software development method and practice. The book brings together some 75 short pieces about principles and techniques for requirements analysis, specification and design. The ideas discussed are deep, but at the same time lightly and wittily expressed. The book is fun to read, rewarding the reader with many valuable and novel insights. Some sacred cows, including top-down development, dataflow diagrams and the distinction between What and How, are led to the slaughter. Readers will be provoked—perhaps to fury, perhaps to enthusiasm, but surely to think more deeply about topics and issues of central importance in the field of software development. There are new ideas about problem structuring, based on the concept of a problem frame, leading to a clearer notion of complexity and how to deal with it. And other important topics include: * Principles for evaluating development methods * New approaches to capturing and describing requirements and specifications, based on the relationship between the software system and the problem context * The technology of description in software, including new ideas such as designations, the separation of descriptive moods and the scope and span of description * Incisive information about the proper role of mathematics and formalism.

My Personal Review:
Each of Jackson's books has expanded on the previous one. PRINCIPLES OF PROGRAM DESIGN was about program design. SOFTWARE DEVELOPMENT was about systems development (specifically, for information systems). Now, SRS takes it further, and considers techniques for specifying different kinds of systems, and systems that require multiple viewpoints to specify them.
One of the main points of the book is to promote the idea that many systems must operate in multiple problem domains and so require multiple techniques in their specification. (Does your information system also have security/access requirements in addition to its functional requirements? Then you have a multi-frame problem, and you should read this book.) The second major emphasis of the book is on describing the application domain -- the real world where the systems users conduct business -- before embarking on designing the system. That is, on understanding the problem before you jump into trying to solve it. If you've ever worked on a huge system development project that turned out to be a catastrophe because halfway through the project you discovered that there was a whole area of user requirements that you didn't even know about, then you should read this book. Find out what went wrong and get some ideas about how to avoid it in the future.

This is an excellent book about specifying computer systems, and I can recommend it to anyone. It is especially highly recommended if you enjoy thinking about software development at a fairly high, abstract level (a level above the one where you are designing code and database structures).

For More 5 Star Customer Reviews and Lowest Price: