

Download A Philosophy Of Software Design

The book first introduces the fundamental problem in software design, which is managing complexity. It then discusses philosophical issues about how to approach the software design process, and it presents a collection of design principles to apply during software design. The book also introduces a set of red flags that identify design problems. You can apply the ideas in this book to minimize the complexity of large software systems, so that you can write software more quickly and cheaply. A Philosophy takes a look at complexity in software, and wants you "to use complexity to guide the design of software through its lifetime." The author ran an undergraduate course on software design, modeled after the approach to teaching writing essays (draft, write, critique, rewrite, critique, rewrite again), and used that experience, combined with a long career of developing many large systems, to develop categories of complexity and mitigations. I'm trying to read all the good writing about software design. This is very easy because not very much has been written: it turns out that it's much easier to write an article about how to write a Tetris AI as a containerized Kotlin microservice than it is to shed insight on how to write good code. And so, when I heard about John Ousterhout's new book "A Philosophy of Software Design," I ordered my copy immediately. A Philosophy of Software Design is a good book, but not a great one. But it's one of the only books of its kind out there. Hence, I am proud to recommend it to all junior-level software engineers. But it's one of the only books of its kind out there.