

## Foreword

The first time I met Dr. Carola Lilienthal was at a conference on Domain-Driven Design where we were both presenting. As she went through her slides and discussed many of the concepts in this book, two things became clear.

First, her understanding of software architecture is simultaneously deep and broad. The advice she gives comes from decades of hands-on experience working with hundreds of legacy systems. She works in the trenches, far removed from the ivory tower that so many software architects move into. Yet, technical as she is, her expertise transcends a single specialty. Her stories, examples, and advice apply to a wide range of domains, languages, and technical stacks. While it might be rare to find an individual with equal breadth and depth, it's exactly the skill set that's required for a book on making software architecture sustainable.

The second thing I noticed was the way she lit up when she talked. For her, making software systems more modular, extensible, and sustainable isn't simply an intellectual or academic pursuit. It's a passion. The joy, hope, and optimism she exudes when talking about improving software architecture is contagious. While this might seem trivial on the surface, make no mistake—that mentality is a crucial asset when you're wading through the mire and chaos of a system bogged down with technical debt. So many systems are abandoned because their maintainers lacked not just the skill to make a change, but the hope that a change was even possible.

For these reasons, I'm thrilled that Dr. Lilienthal has decided to share her experience with the world through her book, *Sustainable Software Architecture*. Legacy code is notoriously difficult to work with for a variety of reasons, not the least of which is the amount of interdependency that exists within a system. Throughout the book, Dr. Lilienthal has provided sound advice on diagnosing, understanding, disentangling, and ultimately preventing the issues that make

software systems brittle and subject to breakage. In addition to the technical examples that you'd expect in a book on software architecture, she takes the time to dive into the behavioral and human aspects that impact sustainability and, in my experience, are inextricably linked to the health of a codebase. She also expertly zooms out, exploring architecture concepts such as domains and layers, and then zooms in to the class level where your typical developer works day-to-day. This holistic approach is crucial for implementing long-lasting change.

There is an immense amount of gratification to be found in making a system sustainable. Too often in our society, “makers” (people who get the most joy during the initial phases of a project) are the ones who enjoy all the glory. However, it's important to recognize the tremendous value that already resides within existing software systems, along with the value of “menders” like Dr. Lilienthal who are eager to dive in and make these systems better. Legacy code runs our world. While it might be tempting to bulldoze a system and start over, doing so, especially if done in haste, rarely accomplishes the goal that we ultimately want—software systems that are easier to modify, extend, grow, and maintain. The better path, the path that ultimately uses fewer resources, causes less frustration, and gets better results, is the path of committed sustainability, and taking the first step down that path is just a few page turns away.

Andrea Goulet  
CEO, Corgibytes  
Founder, Legacy Code Rocks