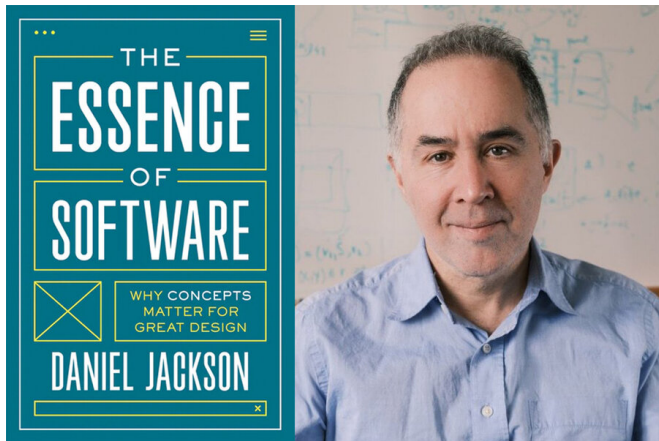


# Book explores conceptual clarity and a new theory of software design

1 December 2021, by Rachel Gordon



MIT Professor Daniel Jackson, who is the associate director of the Computer Science and Artificial Intelligence Laboratory (CSAIL), examines the design and implementation of software through a concept-based approach in his new book, "The Essence of Software." Credit: CSAIL

Sometimes, software is just like us. It can be bloated, slow, and messy. Humans might see a doctor if these symptoms persist (maybe not for messiness), but rarely do we push a flawed software program to go see its developer time and time again.

The answer to why our [software](#) is flawed is ensnared in a web of reliance on flashy hardware, limits of a "code-and-fix" approach, and inadequate design. MIT Professor Daniel Jackson, who is the associate director of MIT's Computer Science and Artificial Intelligence Laboratory (CSAIL), looked at the existing limitations to create a new framework to improve how our programs function. His theory of software design takes a human-centered approach that views an app as a collection of interacting concepts. "The Essence of Software," Jackson's new book, draws on his many years of software research, including designing Alloy, an

open source language and analyzer for software modeling.

**Q: Bugs. Security flaws. Design flaws. Has software always been bad?**

A: Software is actually better than it ever was. It's just that the power and functionality of software has grown so rapidly that we haven't always been able to keep up. And there are some software products (Apple Keynote, for example) that are close to perfect—easy to use, flexible, with almost no bugs. My book offers an approach that will empower everyone to make software that good.

**Q: In your new book, "[The Essence of Software](#)," you introduce a theory of software design that demonstrates how a software system "can be viewed as a collection of interacting concepts." How does this overturn conventional wisdom?**

A: First, [conventional wisdom](#) sees [user experience](#) primarily in the [user interface](#)—its layout, colors, labels, etc. Concept design goes deeper, to address the fundamental mechanisms that the programmer builds and the user experiences.

Second, most apps have large areas of overlapping functionality, but existing approaches don't recognize that, and developers build the same pieces of functionality again and again as if they were new, without taking advantage of the fact they've been built many times before. Just think about how many social media apps have implemented up-voting or comments or favorites, for example. Concepts let you identify these reuse opportunities and take advantage of accumulated design wisdom.

**Q: The year 2021 was one of the worst years for data breaches. Boeing 787s have to be rebooted every 51 minutes to prevent "several potentially catastrophic failure scenarios." Can your approach help with these kinds of security**

## and safety issues?

A: A high proportion of security and safety issues come from a lack of clarity in the design. Concepts can help with that. More directly, concepts can ensure that users actually understand the effects of their actions, and we know that many disasters happen because users do the wrong thing. In the area of security, getting the user to do the wrong thing (such as granting access to someone who shouldn't have access) is usually the easiest path to taking control of a system. So, if you can [design](#) an app to make it harder for users to do things they'll regret, you can mitigate this problem.

*This story is republished courtesy of MIT News ([web.mit.edu/newsoffice/](http://web.mit.edu/newsoffice/)), a popular site that covers news about MIT research, innovation and teaching.*

Provided by Massachusetts Institute of Technology

APA citation: Book explores conceptual clarity and a new theory of software design (2021, December 1) retrieved 26 May 2022 from <https://techxplore.com/news/2021-12-explores-clarity-theory-software.html>

*This document is subject to copyright. Apart from any fair dealing for the purpose of private study or research, no part may be reproduced without the written permission. The content is provided for information purposes only.*