

Review

Search

**10 things software developers should learn about learning**

Brown N., Hermans F., Margulieux L. *Communications of the ACM* 67 78-87, 2024. Type: Article

Date Reviewed: Feb 22 2024

[Full Text](#)

As software developers, we understand the detailed workings of the different components of our computer systems. And--probably due to how computers were presented since their appearance as "digital brains" in the 1940s--we sometimes believe we can transpose that knowledge to how our biological brains work, be it as learners or as problem solvers. This article aims at making the reader understand several mechanisms related to how learning and problem solving actually work in our brains. It focuses on helping expert developers convey knowledge to new learners, as well as learners who need to get up to speed and "start coding." The article's narrative revolves around software developers, but much of what it presents can be applied to different problem domains.

The article takes this mission through ten points, with roughly the same space given to each of them, starting with wrong assumptions many people have about the similarities between computers and our brains. The first section, "Human Memory Is Not Made of Bits," explains the brain processes of remembering as a way of strengthening the force of a memory ("reconsolidation") and the role of activation in related network pathways. The second section, "Human Memory Is Composed of One Limited and One Unlimited System," goes on to explain the organization of memories in the brain between long-term memory (functionally limitless, permanent storage) and working memory (storing little amounts of information used for solving a problem at hand). However, the focus soon shifts to how experience in knowledge leads to different ways of using the same concepts, the importance of going from abstract to concrete knowledge applications and back, and the role of skills repetition over time.

Toward the end of the article, the focus shifts from the mechanical act of learning to expertise. Section 6, "The Internet Has Not Made Learning Obsolete," emphasizes that problem solving is not just putting together the pieces of a puzzle; searching online for solutions to a problem does not activate the neural pathways that would get fired up otherwise. The final sections tackle the differences that expertise brings to play when teaching or training a newcomer: the same tools that help the beginner's productivity as "training wheels" will often hamper the expert user's as their knowledge has become automated.

The article is written with a very informal and easy-to-read tone and vocabulary, and brings forward several issues that might seem like commonsense but do ring bells when it comes to my own experiences both as a software developer and as a teacher. The article closes by suggesting several books that further expand on the issues it brings forward. While I could not identify a single focus or thesis with which to characterize this article, the several points it makes will likely help readers better understand (and bring forward to consciousness) mental processes often taken for granted, and consider often-overlooked aspects when transmitting knowledge to newcomers.

Reviewer: [Gunnar Wolf](#)

Review #: CR147713

[+](#) SHARE [f](#) [t](#) [e](#) ...

Would you recommend this review?  yes  no [Enter](#)

Other reviews under "Software Development":

	Date
<a href="#">Strategies for software engineering</a> Ould M., John Wiley & Sons, Inc., New York, NY, 1990. Type: Book (9780471926283)	Oct 1 1991
<a href="#">Applications strategies for risk analysis</a> Charette R., Intertext Pubs./McGraw-Hill Book Co., New York, NY, 1990. Type: Book (9780070108882)	Aug 1 1992
<a href="#">A survey of exploratory software development</a> Trenouth J. <i>The Computer Journal</i> 34(2): 153-163, 1991. Type: Article	Nov 1 1991

[more...](#)

[E-Mail This](#)

[Printer-Friendly](#)