# 11 Self-Taught Software Engineers That Have Made An Impact

Besides the most well-known and famous software engineers (namely, Bill Gates and Elon Musk), several self-taught and self-made software engineers have made an impact on how software is used and developed in the tech industry.

## 1. Ada Lovelace

Considered one of the first computer science engineers, laying the foundation of what we now know as software engineering, Ada Lovelace was also the first tech visionary of her time. Born in 19th century London, Lovelace was the daughter of the famed romantic poet Lord Byron but was raised by her mother, who removed Lovelace from the influence of her father, whom she never met, requiring a logic-minded education of her daughter.

Ada, as a result, was afforded a more advanced technical education than most girls her age, yet was taught at a different level than men in the scientific and mathematical fields. Even so, she rose above what was often expected for women of her station. Teaming up with another brilliant mathematician, professor Charles Baggage, Lovelace assisted in creating one of the first computation machines, the Analytical Engine.

While not technically defined as a software engineer by modern standards, we wouldn't have the level of computation in machinery that we do today without Ada Lovelace and her gifted, mathematically poetic mind.

## 2. Charles Baggage

Born in 18th century England, Baggage was the son of a banker and was self-taught in mathematics as a child, though certainly having numbers run in the family was a help. Continuing as a child prodigy, he was far more advanced than teachers in mathematics, founding an analytical society that significantly improved the department.

As a mathematician in the profession, Baggage was deeply interested in analytical computation, laying the foundation of software engineering with his young protege and eventual partner on equal footing and even of more advanced capabilities, the aforementioned Ada Lovelace. He invented the basis of the Difference Engine, later advancing onto the Analytical Engine with Lovelace. Considered one of the first computers ever, despite being inherently flawed and never a fully-fledged programming success, the machine laid down the foundation of what analytics could do in mechanized form.

## 3. The Scheutz Brothers

While often overlooked in software engineering and history in general, Georg and his son (not, as one would assume, his brother) Edvard Scheutz were impartial to the design and build of Lovelace and Baggage's Analytical Engine. Not trained or educated as mathematicians or scientists, the brothers embody the entrepreneurial and driven spirit of modern self-taught engineers.

Printer by trade, George Scheutzdabbled in other fields beyond publishing. After helping build what he knew would be a world-changing machine, he built other such machines commercially, selling the Stockholm engine with his teenage son, Edvard. Unfortunately, neither made much of a profit from likewise inventions, leading to their eventual financial ruin but imperative in the historical lexicon of self-taught software engineers and inventors.

## 4. Alan Turing

Speaking of Charles Baggage, the Charles Baggage Institute, College of Science and Engineering, at the University of Minnesota, offers a publicly transcribed interview in which the intertwined mathematical theories of the great Alan Turing and John Von Neumann is discussed. Turing, a gifted cryptographer in World War II, was later convicted as a criminal at a time when his brilliant mind was tragically overlooked for the "crime" of homosexuality.

While later proving his salt as a mathematician and one of the early founders of software engineering, Turing didn't do well in school. He got bad grades and aggravated his teachers. He was truly a self-made man, assisting Allies in the defeat of Axis powers in WWII by cracking the German "Enigma" coding machine.

## 3. John von Neumann

Like Turing, John von Neumann was born in the early 20th century, moving from Europe in the 1930s to teach at Princeton University in the United States. He trained as a mathematician in Budapest, Hungary, at a time when being of Jewish descent earned the ire and discrimination of his peers, and later studied both chemistry and mathematics before being invited to lecture in the U.S.

von Neumann was one of the self-taught mathematicians who worked on advancing the electrical components of computation machines, applying analytical theory to research the vast scope of what these machines could one day achieve, including hypothetical theories on how to engineer the software capabilities (particularly the mini-max theorem and von Neumann-Morgenstern theory) that we now use daily.

## 6. Frances “Fran” Allen

Inspired by her math teacher in high school, Allen pursued a teacher’s degree in mathematics at university, but after teaching at the secondary level, realized she’d need to continue to a master’s degree for her full teacher’s license. Allen decided, instead, to take a course in IBM programming, later taking on a job at IBM to pay off student debt and go back to public school teaching. Instead, she worked for the company for another 45 years, teaching herself and others how to advance further IBM’s pivotal role in optimizing software transformations.

Allen was the first woman to become an IBM Fellow in 1989, later being awarded the prestigious ACM Turing Award in 2006.

## 7. Margaret Hamilton

How can we write about self-taught software engineers without including Margaret Hamilton? Born in the late 1930s in small-town Indiana, Hamilton is another mathematician whose specialty moved not from teaching arithmetic in school, as many women at the time were assumed to do with a math degree, but to making landmark advances in software development.

Hamilton not only analyzed theories and wrote about them -- she sent a rocket to the *moon.* Leading a team of software engineers at MIT, Hamilton developed the code that historically launched Apollo 11 into space in 1961, landing humanity on the moon for the first time.

## 8. Edsger W. Dijkstra

Dijkstra, a theoretical physicist turned software engineer, worked as a programmer at Mathematisch Centrum’s Computation Department for three years at Leiden University, eventually combining the two fields, among others, in a way that altered the path of software engineering forever.

Self-described as not suited to software engineering, specifically in programming, Dijkstra nonetheless taught himself how to look at his role in computer engineering in a novel light. He invented an algorithm that would pave the way to directional calculations: how to find the shortest path from one destination to the other. His work won him numerous awards and achievements (including the ACM Turing Award) throughout his lifetime as he continued to invent, research and philosophize about the intricacies of mathematics, analytics and programming.

## 9. Grace Hooper

Another great female mind apart of influential software engineers, Hooper, expressed interest in engineering and mathematics at a young age. While Hooper earned a Masters and PhD in Mathematics from Yale University, she left her post-teaching at the local college and joined the Navy in 1943. Having a mind for math and engineering, Hooper was recruited into the Bureau of Ordnance Computation Project at Harvard University during the second world war, working on a calculator (Mark I) that would later be recognized as an early prototype of an electronic computer.

After the war, Hooper continued to work in the Navy in programming and software languages. She earned both recognition for her distinguished career in the Navy and her role in advancing computer science. In 1987 she was awarded the Defense Distinguished Service Medal, then was recognized as a distinguished fellow of the British Computer Society in 1973.

## 10. Donald Knuth

Another recipient of the prestigious ACM Turing Award, among other accommodations, Knuth is known as the "Yoda of Silicon Valley." A captivating and gifted storyteller and computer programmer, he is renowned for his ability to write and speak on the world of software engineering in a way that is accessible and interesting to the everyday person.

Beginning in 1962, Knuth began writing (and is still writing) a series of books well-known and beloved throughout Silicone Valley; *The Art of Computer Programming* (or TAOCP for short), which is still widely read by accomplished and aspiring software engineers.

## 11. John Carmack

Like Knutch, Carmack is one of the more modern examples of self-taught software engineers. Growing up in middle America in the 1970s and 80s, Carmack lived and breathed gaming culture. Like other tech giants of Silicone Valley fame, Carmack opted to drop out of the University of Missouri and freelance in programming instead, finding his niche in programming video games.

Notably, Carmack revolutionized gaming algorithms, giving way to classic games like the *Doom* and *Wolfenstein* series, among others. He continues to research other avenues of programming, such as AI projects and is one of the successful examples of modern self-taught software engineers.

Linked Sources:

Ada Lovelace
https://www.newyorker.com/tech/annals-of-technology/ada-lovelace-the-first-tech-visionary
https://www.sdsc.edu/ScienceWomen/lovelace.html

Charles Baggage
https://cse.umn.edu/cbi/who-was-charles-babbage
https://www.britannica.com/technology/Analytical-Engine

The Scheutz Brothers
https://www.computerhistory.org/babbage/georgedvardscheutz/

Alan Turning
https://cse.umn.edu/cbi
https://conservancy.umn.edu/handle/11299/107493
https://www.pbs.org/newshour/science/8-things-didnt-know-alan-turing
https://www.washingtonpost.com/national/health-science/what-imitation-game-didnt-tell-you-about-alan-turings-greatest-triumph/2015/02/20/ffd210b6-b606-11e4-9423-f3d0a1ec335c_story.html

John von Neumann
https://mathshistory.st-andrews.ac.uk/Biographies/Von_Neumann/
https://www.britannica.com/science/mini-max-theorem
https://www.britannica.com/science/game-theory/The-von-Neumann-Morgenstern-theory#ref22625

Fran Allen
https://www.computer.org/profiles/frances-allen/
https://amturing.acm.org/award_winners/allen_1012327.cfm
https://scientificwomen.net/women/allen-frances-108

Margaret Hamilton
https://news.mit.edu/2016/scene-at-mit-margaret-hamilton-apollo-code-0817

https://www.theguardian.com/technology/2019/jul/13/margaret-hamilton-computer-scientist-interview-software-apollo-missions-1969-moon-landing-nasa-women

Edsger W. Dijkstra
https://www.cwi.nl/about/history/e-w-dijkstra-brilliant-colourful-and-opinionated
https://www.cs.utexas.edu/users/EWD/

Grace Hooper
https://www.womenshistory.org/education-resources/biographies/grace-hopper
https://news.yale.edu/2017/02/10/grace-murray-hopper-1906-1992-legacy-innovation-and-service
Donald Knuth
https://www.openculture.com/2019/01/110-lectures-by-donald-knuth.html
https://www.quantamagazine.org/computer-scientist-donald-knuth-cant-stop-telling-stories-20200416/

John Carmack
https://academickids.com/encyclopedia/index.php/John_Carmack
https://techcrunch.com/2019/11/13/john-carmack-steps-down-at-oculus-to-pursue-ai-passion-project-before-i-get-too-old/