Influential Men and Women of Software

Kathy Gurchiek
CrossTalk Managing Editor

Looking at the Evolution of Software would not be complete without a nod to those software heavyweights whose contributions furthered software development. Below, CrossTalk *recognizes some of those people.*

Charles Babbage

Inventor in the early 1800s of the difference engine and design for the analytical engine. He was a well-educated young man in Great Britain who wanted to be a mathematician and scientist. One evening in 1812 while studying a table of logarithms, he realized calculating them could be automated and he began to design the Analytical Engine for that purpose. Like projects since, it was never completed due to cost overruns and reluctance by the British to provide additional funding.

Lady Ada Augusta Byron

Countess of Lovelace and daughter of poet Lord Byron, she was a 19th century mathematician who has unofficially been called the first computer programmer. Through lengthy correspondence and notes during the 1800s, she interpreted Babbage's "thinking machine," and was able to foresee, and describe in simple terms, the symbolic processing by machine.

Alan Kay

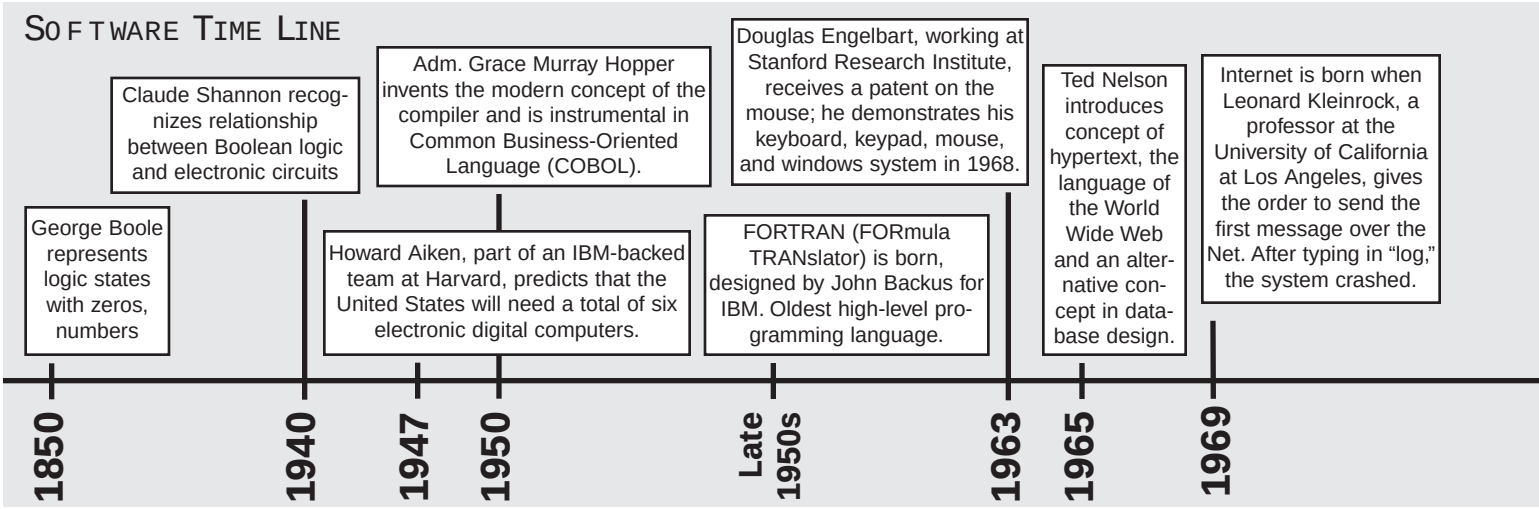
Credited with coining the term "object-oriented" (OO) programming language, a term which came into use in 1970, four years after the first OO programming language (Simula) was introduced.

"It would appear that we have reached the limits of what it is possible to achieve with computer technology, although one should be careful with such statements, as they tend to sound pretty silly in five years."
— John Von Neumann, 1949

Alan Turing

In the 1930s he developed what is now known as the Turing Machine, a theoretical computer that could "compute all that is computable" using limited instructions and infinite working storage. Using it as standard, two Italian mathematicians proved that any programming language needed only the sequence, the decision statement, and the iterative looping structure to implement any computable algorithm.

Software Time Line



Capers Jones

Known for work in the quantification of software productivity and quality, and in developing estimation and measurement tools, which use his quantification methods. In the early 1970s, he noticed there was a consistent bias in software project data. The discovery of bias patterns in lines of code metric confirmed the economic validity of Allan Albrecht's function point metric. His book, *Programming Productivity*, was the first to quantify the paradox of lines of code metrics and show the bias by language level.

Watts S. Humphrey

Research scientist for the Software Process Program he founded as part of the Software Engineering Institute located in Pittsburgh. He is the author of *A Discipline for Software Engineering* (1995), *Managing Technical People* (1996), and *Introduction to the Personal Software Process* (1997).

Barry W. Boehm

While employed by TRW Corp., Boehm wrote *Software Engineering Economics.* Published in 1981, it completely described the Constructive Cost Model (COCOMO) used in software cost and schedule estimation. With his book, this model quickly gained popularity and user's groups emerged throughout the world.

Tim Berners-Lee

Inventor of the World Wide Web, which runs on the Internet, and hailed by *Time* magazine as one of the 100 greatest minds of this century. The web made the Internet useable and useful to all types of people and applications. Before the web was born, the Internet was used mostly by scientists and the military and one had to be a progammer to make use of it. He is the director of the World Wide Web Consortium at the MIT Laboratory for Computer Science.

"Technology makes it possible for people to gain control over everything, except technology."
— John Tudor

Edward Yourdon

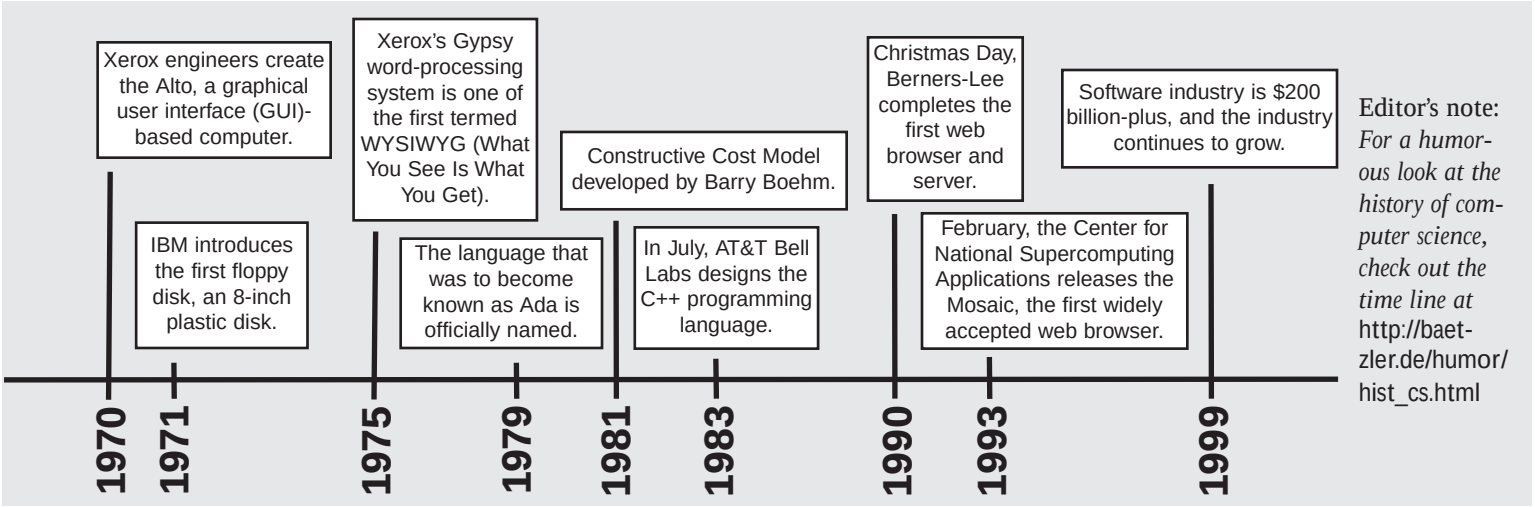
The publisher of *American Programmer* is widely known as the developer of the method of structured systems analysis and design.

Adm. "Amazing" Grace Murray Hopper

Creator of Common Business Oriented Language (COBOL). She was an officer in the Navy who became an Admiral. COBOL came about in the 1950s when the need for higher order languages was seen as a way to increase the productivity of programming computer applications.

References:

1. Marciniak, John J., Editor-in-Chief, *Encyclopedia of Software Engineering*, John Wiley & Sons Inc., New York, 1994, pages 25, 36, 103, 408, 605, 688-89, 691, 721.
2. World Wide Web History Project, http://www.webhistory.org/historyday/abstracts.html
3. Humphrey, Watts, *CrossTalk: The Journal of Defense Software Engineering*, February 1999
4. Keyes, Jessica, *Software Engineering Productivity Handbook*, McGraw Hill, 1992, pp. 80, 116, 332.
5. A Brief History of the Computer, http://www.qvctc.commnet.edu/classes/csc277/timeline.html



Editor's note: *For a humorous look at the history of computer science, check out the time line at* http://baetzler.de/humor/hist_cs.html