

John Backus

John Backus was an American Computer Scientist credited with a wide range of innovations such as the development of FORTRAN, Backus-Naur Form (BNF) and also his research into function-level programming. Clearly, as an individual, his impact on the development of software was immense.

FORTRAN

John Backus led the IBM team that developed FORTRAN, often referred to as the first widely used high-level language. FORTRAN became known for its fast computation and efficiency, and for decades dominated scientific and engineering applications such as weather prediction, computational physics and fluid dynamics. Its performance meant that it became a benchmark for the fastest supercomputers. FORTRAN's design also influenced many subsequent languages such as BASIC. Ken Thompson, the creator of Unix, even remarked that *"95 percent of the people who programmed in the early years would never have done it without Fortran."*

Backus felt frustrated by his experience of *"hand-to-hand combat with the machine"* and desired to simplify programming. The goal in developing FORTRAN was to create a language more readable by humans, since assembly language was so tedious. In 1953, he wrote a memo to his boss requesting he lead a research project in order to accomplish this. His proposal was approved and he began hiring a team of programmers and mathematicians.

The IBM team's challenge was not the design of the language, but rather the translation of it into machine code. In 1954, Backus and his team were confident enough in their research to publish a paper, called "Preliminary Report, Specifications for the IBM Mathematical FORMula TRANslating System, FORTRAN." Originally prospected to take 6 months, it ended up taking 2 years, likely due to the lack of even any software engineering practices at the time. As Backus himself remarked:

"We did not know what we wanted and how to do it. It just sort of grew. The first struggle was over what the language would look like. Then how to parse expressions - it was a big problem and what we did looks astonishingly clumsy now...."

Eventually the project was completed, containing 25,000 lines of machine code. A worthy undertaking surely, that would severely reduce the need for anybody else to have to crawl through an appalling mess of assembly.

Backus' leading of the team was apparently very informal. He ignored IBM's yearly performance reviews, as he deemed them ill-suited for programmers. According to himself: *"Much of my work has come from being lazy"*. Referring to his earlier work, he also stated, *"I didn't like writing programs, and so, when I was working on the IBM 701, writing programs for computing missile trajectories, I started work on a programming system to make it easier to write programs for the 701. And that wound up as something called Speedcoding."* His disdain for arduous process is not surprising, considering his difficulties and unenjoyment experienced during his education, having been expelled from University due to lack of attendance. Regardless, his leaning towards laziness managed to make it easier for every programmer which continues to write in the aftermath of FORTRAN.

Backus-Naur Form

Another widely used development accredited to Backus is Backus-Naur Form, or BNF. Following development of the FORTRAN compiler, it's not surprising that he further sought to assist the compiler creation process. Together with Peter Naur, Backus created a formal and unambiguous notation for the description of the syntax of higher level languages.

Forms of BNF are implemented in many compiler creation tools such as GNU Bison, the GNU version of Yacc. BNF is even used outside compiler software such as communication protocols and document formats. BNF led to him being awarded the Turing award in 1977.

Function-level Programming

Despite essentially starting the high-level programming trend, Backus was not a fan of subsequent languages, describing them as too complex and inflexible: *"it takes pages and pages of gobbledygook to describe how a programming language works, it's hard to prove that a given program actually does what it is supposed to."*

Backus, however, went on to attempt to develop another language, FP, which he described as "function-level". He published a paper called "Can Programming be Liberated from the von Neumann Style?", which did not actually do much to promote his language. His research sparked interest in functional programming languages, rather than the function-level programming paradigm he was trying to promote. Very few implementations of FP are around. He also later went on to develop FL, a successor to FP, but it remained a research project.

Without the work of Backus with high-level languages, large scale projects could never have been remotely conceived. Although widely considered old fashioned, FORTRAN is still in use today in many contexts. BNF will also continue to be used extensively. Without a doubt, Backus has made a massive impact on software engineering.

1. Sonia Weiss. Short Biography on John Backus' life.
http://www.thocp.net/biographies/backus_john.htm
2. IBM Archive. Concerning John Backus and FORTRAN.
http://www-03.ibm.com/ibm/history/exhibits/builders/builders_backus2.html
3. J J O'Connor and E F Robertson. Short biographical piece.
<https://web.archive.org/web/20030605163136/http://www-gap.dcs.st-and.ac.uk/~history/Mathematicians/Backus.html>
4. Steve Lohr. Obituary for John W. Backus.
<https://www.nytimes.com/2007/03/20/business/20backus.html>