

# Edsger W. Dijkstra

## **Introduction:**

Edsger W. Dijkstra was an early pioneer in computer science and developed many computer science and computer engineering fundamentals. He was a computer scientist, software engineer, programmer, and a mathematician. I chose to write this biography about him because I admire his extensive talent in multiple different topics, his humbleness and calmness, and his simple yet effective methodical approach of elegance in his work.

## **Background:**

Edsger W. Dijkstra was born in Rotterdam in 1930 to a family with an intellectual background. His father was a chemistry teacher, and his mother was a mathematician. He studied theoretical physics in the university of Leiden and later got a PHD in computing science in the university of Amsterdam. He sadly died of cancer in 2002 at the age of 72.

## **His Work:**

### **Algorithms:**

Dijkstra's algorithmic work had a profound impact on the field of computer science and software engineering. He is most famous for his simple but effective algorithm use for finding the shortest path between two nodes in a connected tree. What I find fascinating is that he supposedly designed this algorithm in 20 minutes in car. This algorithm now has a vast number of use cases, such as in routing protocols that power the internet, and is now even used in google maps to calculate the shortest route between two destinations. It can be useful in many data structures that are used by software engineers to write their software and solve various types of problems.

### **Programming and software engineering:**

Dijkstra is also accredited for the adoption of structured programming as the preferred way of software engineering. He realised that as programs get bigger, and more complex, the only way to successfully design them would be in a more rational and structured way. Before 1968, goto statements were the norm in programming. Dijkstra was a critic of goto statements because he believed that writing programs and software would be much easier and logical without goto statements and rather in a more structured way. He outlined his ideas about this in one of his papers called "a case against the goto statement". He was quite vocal about these ideas that eventually goto statements were nearly eliminated from programming and replaced with more structured methodologies such as for and while statements. His ideas about structured programming laid the foundation of what programming and software engineering is today and are the basis for object-oriented programming which some programming languages are build around. He also came up with the phrase "two or more, use a for" which is a general rule for programmers and software engineers to restructure and encapsulate their code in a loop once it gets messy.

### **Concurrency, X8, Distributed Systems:**

Dijkstra wrote the operating system known as “THE” and software for the Electrologica computer. This computer was the biggest in Holland at the time. “THE” was one of the first multiprogramming systems. This concept of multiprogramming has revolutionised the area of computer engineering as it demonstrated that it is possible to run multiple pieces of software on one computer at a time. Around this time, he wrote a paper that first introduced the concepts of concurrent systems such as the famous mutual exclusion problem and solution, critical sections, deadlock, synchronisation, and semaphores which are also known as locks, and showed how to deal with the challenges that arise with concurrency. I have used these useful concepts in my own personal multiprocessing software engineering projects, and it amazes me how this one man thought of most of it himself. This also ties in closely with his work on distributed systems, which is a way of how many computers can work together to complete a common task. Some people even argue that he started the field entirely on his own with one of his papers on self-stabilising systems. It has opened a whole new type of software engineering, which is software engineering for distributed systems. In my opinion, his work in this area changed the world today because most of the things that we do online, is facilitated by an underlying data centre or web server working behind the scenes, made up of many computers that work together using the concepts of concurrency and distributed systems.

### **Style:**

Edsger W.Dijkstra had a unique style and personality. He is known for valuing elegance and simplicity while solving problems in software engineering and other related computer science fields. His famous Dijkstra’s algorithm is the prime example of this ingenious simplicity and elegance. He once said that “quality, correctness and elegance should characterise a computer program”. I cannot agree more with him on this because computer programs and software can get extremely complicated as they grow so it is essential to keep things as simple as possible for the task from the very start. This is especially important when working on a large software engineering project with numerous other people because the simpler a piece of code is, the faster they will understand it and will be able to work from it. In my opinion, most programmers nowadays strive for this “quality, correctness and elegance” in their work. Another resounding quote by him is “simplicity is the prerequisite for reliability”. I also completely agree with this quote. From my own experience, I found that if the code for a solution or a program is simple, it leads to less bugs and more reliability. This is because it is much easier to find those bugs if a program is simple and short, similarly to how it is easier to find a person among a crowd if the crowd is small. In my opinion, people sometimes try to show off by making their work look very complicated and sophisticated, in most disciplines. However, there is no place for this complication of things in the world of software engineering as it leads to less efficient, less correct programs. Edsger knew this and made sure everyone else did too.

### **Closing thoughts:**

In my opinion, Edsger W.Dijkstra had one of the largest impacts on the area of computer science ever. His work contributed to many different areas of computer science and software engineering. I would go as far to say that almost anything we do that is related to

computers or IT had some of their concepts and fundamentals developed by Edsger in one way or another. I admire him because of his many exceptional talents in a lot of different areas.

**References:** <https://www.computer.org/profiles/edsger-dijkstra>

<https://history.computer.org/pioneers/dijkstra.html>

<https://www.cs.utexas.edu/~EWD/>

<https://www.britannica.com/biography/Edsger-Dijkstra>

[https://en.wikipedia.org/wiki/Edsger\\_W.\\_Dijkstra#Pioneering\\_contributions\\_and\\_impact\\_on\\_computing\\_science](https://en.wikipedia.org/wiki/Edsger_W._Dijkstra#Pioneering_contributions_and_impact_on_computing_science)

[https://amturing.acm.org/award\\_winners/dijkstra\\_1053701.cfm](https://amturing.acm.org/award_winners/dijkstra_1053701.cfm)

<https://www.cwi.nl/about/history/e-w-dijkstra-brilliant-colourful-and-opinionated>

[https://www.goodreads.com/author/quotes/1013817.Edsger\\_W\\_Dijkstra](https://www.goodreads.com/author/quotes/1013817.Edsger_W_Dijkstra)