CS3012

Software Engineering

Biography of an influential software engineer

Donald Knuth, PhD

Jack Donal Collins

Computer Science & Business

Junior Sophister

16323107

Collinj6@tcd.ie

Introduction

Knuth is widely considered as one of the most influential software engineers of the 20th century and is credited as being the *"father of the analysis of algorithms"* (Donald Knuth Brief Biography, 2018). Knuth's most famous book, The Art of Computer Programming is so widely renowned that Bill gates once jokingly offered a job to anyone who had completed the entire volume. In 2018 Knuth celebrates his 80th year, his success is so widely renowned that a series of events and celebrations entitled Knuth80 have taken place to celebrate his achievements. Knuth's wealth of achievement and perseverance is an inspiration which one would do well to try and replicate. His work has touched so many different spheres, from the approach engineers take, to the technology academics use to publish their research and even the classes students who study computer science take.

Early Life & Education

Knuth was born in Milwaukee, Wisconsin in January 1938 (Robertson, 2015). He attended the Lutheran school in which his father worked. It is said that the formal and rigid education taught at Lutheran schools is what gave Knuth his love of structure. Lutheran schools are well known for the emphasis they place on English grammar, the precursor to Knuth's fascination of computational thinking was his lauding over the different ways to construct a sentence. His love of mathematics did not manifest itself in High School which was predominately dominated by his love of music. That being said he did graduate from his high school with the highest GPA ever recorded for the school. (Robertson, 2015)

Education

In 1960 Knuth received both his bachelors and master's degree in mathematics from the Case Institute of Technology, his bachelor's dissertation being of such high quality that it was deemed of master's merit which was subsequently conferred upon him. (McGill, 2007) It would be considered rare for a doctoral student in mathematics to publish a paper however in what must be considered an exceptionally rare and unique case Knuth published two papers while completing his undergraduate degree (Robertson, 2015). In 1963, Knuth obtained a PhD in mathematics from the California Institute of Technology. Following his PhD, he became a professor at the same university and began work on the Art of Computer Programming. Four years later in 1968 he published the first volume of the originally planned seven volume series while also joining Sandford University (McGill, 2007).

The Art of Programming (TAOP)

In 1966 Addison-Wesley, Knuth's book publisher, realised that Knuth's three-thousand-page book on compilers had become a much larger piece than had been originally planned. The length, breadth and detail of the works resulted in a seven-part volume being planned (Robertson, 2015).

Knuth's aims for the book, which can be found in its preface is to *"organize and summarize what is known about the fast subject of computer methods and to give it firm mathematical and historical foundations"* and *"show that the connection between computers and mathematics is far deeper and more intimate than these traditional relationships would imply."*. It is fair to say that Knuth aimed to create a bible or lexicon for computer programmers, with the first volume on fundamental algorithms designed to give the reader a concrete understanding of the intrinsic link between mathematics and computing

By the end of 1973, Knuth has published his first three volumes of TAOP, his books included all that was known on incredibly important and even basic areas such as random numbers, arithmetic, sorting and searching and combinatorial algorithms . His work on TAOP is so important that it is referenced consistently by University textbooks. The structure of TAOP is not all that dissimilar to the Introductory to Algorithms course we took last year, highlighting the notion that it is commonly agreed that the techniques and tools found in TAOP are an invaluable asset to any software engineer. It is hard to disagree with the fact that his work has had a large impact on not only my education but the formal education of computer scientists and engineers the world over. Knuth's love for perfection is clear, he has a standing offer to pay $2.56 for each error found in TAOP. His error cheques have become icons of legend and almost status symbols.

Impact of the TAOP - Turing Award

The impact of Knuth's work is so great that he was honoured with the AM Turing award in 1974. The official designation for his award was for "major contributions to analysis of algorithms and the design of programming languages, and in particular for his most significant contributions to the 'art of computer programming'" (Walden, 1974). Turing may have invented the computer, but Knuth most definitely had a hand in defining the processes and techniques engineers use.

Academic Impact

As a professor in Sandford University Knuth supervised 28 PhD students while publishing papers such topics such as concrete mathematics as well as the Sandford Graph base, a platform for computational computing. Knuth

TeX

While Knuth was reviewing porotypes for the second volume of TAOP he was dissatisfied with the proposed typeset. The result of which was his aim to develop a new formal and practical typeset named TeX, the precursor to LaTex, a popular application for formal reporting writing both in academia and industry. The necessity to create a more formal document showed.

Knuth built hooks into TeX so that others could make improvements upon it, with the condition that any new program be given a different name. TeX developed an active and passionate community of users and developer with the result being the creation of tools such as LaTeX, ConTeXt, LuaTeX. Knuth's TeX was an early success story for the free and open-source software movement.

TeX demonstrated to the world the power of open source and it sometimes credited with pioneering mas open source development projects, Knuth's development of TeX would later become useful in his proposition of the Literate programming methodology. Having used LaTeX for projects in the past I found it incredibly interesting to learn about Knuth's contribution which I was not initially aware of before further researching Knuth and his work.

Literate Programming Methodology

Knuth was a strong believer of software engineering process so much that he developed a new programming methodology. The (Knuth, 1984)methodology holds the principle that engineers should think of programs as pieces of literature. His belief was that we should focus more on explaining to humans what the computer is doing rather than simply instructing a computer what to do.

Knuth described a literate programmer as someone who strives "*for a program that is comprehensible because its concepts have been introduced in an order that is best for human*

understanding, using a mixture of formal and informal methods that reinforce each other." (Knuth, 1984). The elegance and almost artistic like description of this methodology encourages engineers to think beyond simply completing the task but asks the engineer to consider the implications of the decisions they make about the flow and interpretation of the program. Interestingly, it is quite easy to find a link between this programming methodology and Knuth's formal mathematical background. Literate programming forces the engineer to convince other engineers of the code's functionality rather than just the compiler of the language he or she writes their code in.

Awards

In addition to Knuth's Turing award, he has been honoured with over 100 other awards including the Grace Murray Hopper Award in 1971 and being conferred as a fellow of the American Academy of Arts and Sciences in 1973. The countless awards and acolodaes Knuth has achieved further my points on his influence over modern computer

Retirement & Legacy

Now retired, Knuth has declared his focus on completing TAOP in what he describes as his most important contribution to society. He continues to offer public lectures on a regular basis but only replies to emails and letters in batches as to do otherwise would interrupt the flow. Knuth's determination, even in his later years to complete his work and continue to progress the area is something I admire profusely, especially for someone who shy about their e=own ability even following an extremely successful academic experience.

Knuth's 4th Volume of TAOP has had to be split into two parts, part a of which has been released and part b which is still in development. There is still hope within the community that Knuth will be able to complete TAOP. Regardless, his legacy is intact as a visionary of software engineering.

Bibliography

Donald Knuth Brief Biography. (2018, October 20). Retrieved from Informs.com: https://www.informs.org/content/view/full/268483

Knuth, D. (1984). *Literate Programming.*

McGill. (2007). *McGill*. Retrieved from https://www.cs.mcgill.ca/~rwest/wikispeedia/wpcd/wp/d/Donald_Knuth.htm

Robertson, J. J. (2015, October). *St Andrew's University - Knuth Biography*. Retrieved from St Andrews University Website: http://www-history.mcs.st-andrews.ac.uk/Biographies/Knuth.html

Walden, D. (1974). *Turing Award* . Retrieved from Turing Award : https://amturing.acm.org/award_winners/knuth_1013846.cfm