Alan Cooper was best known for his book about the essentials of User Interface Design. He championed one of the best and most used interfaces for design, virtual basic.

I find it interesting that he is a dropout but also has a degree from a college. In my experience people either work hard at school or don't, and he seems to be a little bit of both. I have always asked myself which is better, school or self taught in computer science and a lot of people argue with me about it. I can always respect someone who pursued an education and be jealous of those who were successful without it. I really love that he got such a good first job with only two years of school. I wish the world was like that now.

He then started his own company and is self-employed. This is a personal dream of mine. I would love to one day be able to be my own boss and run something cool, maybe an AI company, but I do dont think I have the strength for it.

Cooper is known for his strides in interface design and he believes that the most important aspect is to know your user.

Cooper has always favored Microsoft. I also champion their software. Like Cooper, I think their interface is user friendly enough for an average user but also has functionality for an experienced computer scientist.

His ‘aha’ moment was where he figured out that every user would need their own personal interface. It would be kind of crazy to build and create an interface specifically for each user. It would be equally as crazy to expect each user to need the same things or have the same skill level. So making his program where the user can choose what tools they need and what they want to see makes perfect sense. I would say I am surprised no one thought of it before but technology always has these aha moments and someone needs to be responsible for them.

Cooper worked with Bill Gates to develop his ideas with Microsoft. Microsoft took tripod and changed it to be called Ruby. They added QuickBasic to it and created VB.

Cooper changed the format of specific runtime and compile time, and made it available to be dynamic with third party controls. These code objects could now be connected to runtime. Cooper had always liked how dynamic Microsoft was and he ran with this. He made his code objects and tools dynamic. Not only was the usage dynamic but the reach was too. The code would be downloaded and could extend.

Cooper's language was the first to be linked to a library and also had a front end. As a third party, the grid control was an integral part of the language. This could now be changed on a personal level computer to computer.

Ruby was originally a small language that could only compute a few shell commands. But he connected it with DLLs and the program could now be linked to so many things. He wanted Ruby to be connected with a real language. C was the contender at the time. The little shell language was replaced entirely with Quick Basic. This caused a visual front end that was static and permanent. The dynamic aspects were lost in this process.

Cooper's biggest goal was to focus on the users. His focus was a pleasing and easy user interface instead of an easy job for programmers.

Bridges are not built by engineers, they are built by ironworkers. Software programs are not built by engineers, they are built by programmers.

I think Alan Cooper had some really amazing ideas. I like his hyper focus on user interface. To me, that is the most important thing about my programs. I lavish in the awe of my ignorant friends when I show them thousands of lines of code for a simple cute little project. I think about if it looks good and if it is easy to use. I always ask myself if it has everything needed for a user. Lots of my classes recently have been about the pureness of a few lines of code and what they do. I just don't get any passion out of it. I like making and creating beautiful things for users to experience.