

## CROSSTALK would like to thank NAVAIR for sponsoring this issue.



### What is a Software Engineer?

Software engineers, computer engineers, computer scientists, programmers: how often do you hear people using these titles interchangeably, as if these roles all perform the same job? While, yes, they all do work with computers, that would be the extent of these roles' similarities. Saying these roles are the same is like

lumping master chefs, executive chefs, sous chefs, and short-order cooks together because they can all cook. Software engineers have a very unique set of skills that are quite different from the rest of those roles that 'work with computers.'

Let's start by talking about programmers. I think of programmers from the days where software houses were organized into groups of designers and programmers. When a designer finished their work on a product, they would hand over the design to a programmer, who would then code that design into a specific computer language. This working relationship inferred that programmers were only good at writing software but not designing software. With today's powerful development environments, along with the ever-evolving vernacular of the software industry, programming in this day and age means a lot more than just "slinging" code.

But let's take that old-school definition of the programmer and combine it with the ability to design and inject a heavy dose of computational theory. Now just add a white lab coat and you've got yourself a computer scientist (that's my title - although I have never owned a white lab coat). Computer scientists bring their strength to the table in their ability to model something that needs to be studied in a virtual environment. This comes in really handy when the problem you are studying involves explosives or other things that are hard to analyze in the real world (e.g., river currents, blast fragments, forces from a thrust vectoring nozzle). In my career with the DoD, I spent over a dozen years working on a war-game simulation that the U.S. Navy used to develop military tactics. That war-game simulation contained many models interacting with each other to present a complex environment to the users. For those models to be

able to coordinate with each other, an architecture needed to be established to define and organize the lines of communication. Now we are getting seriously beyond "slinging" code.

As more functionality was added to the simulation, performance bottle-necks began to appear. Now the software team found that they had exceeded the limitations of the hardware they are running their simulation on. "It's not our fault!" they said, "Our system is just what it needs to be. We need faster hardware." Something about 'try a bigger hammer' comes to mind. Enter the computer engineer: this individual would look at the system being used to run the simulation and state: "Of course you are having performance problems, your hardware system is built the wrong way and doesn't support the needs of your simulation." Computer engineers are like computer scientists, but are more focused on the hardware side of computers, rather than the theory. Yes, they can both write software, but buy a computer engineer a couple of drinks, and they start talking about AND-gate this, OR-gate that, and how anything can be done with the right hardware.

Meanwhile, over in the corner, a software engineer recommends a more systemic approach to solve the performance problem. "Let's look at what the users need the simulation to do now and where the simulation will most likely grow in the future. Then we can build a computer system that will meet the simulation's computational needs today, and have the hardware architecture to grow in capability as the simulation grows." The computer scientist and the computer engineer turn and look at the software engineer and say: "You sound like an engineer." And there you have it. The strength of a software engineer is in having been trained to apply engineering to the world of computer science. Just as mechanical, electrical, and civil engineers take a methodical approach to solve the problems in their domains, software engineers use the same engineering principles, but in the world of computers.

### Bradley T. Hodgins

Team Software Process Coach and Instructor,  
Process Resource Team  
Naval Air Systems Command