

Software Development May Drive Future Acquisition Cycle Times*

David Tate, Institute for Defense Analyses (dtate@ida.org)

If you look at the last 30 years of defense acquisition, it turns out that **in general it is not taking us longer to develop and field weapon systems**—with a few exceptions. Those exceptions tend to be our most expensive, complex, and highly visible systems.

Should this be surprising? Cost estimators understand that system cost is largely driven by the content of the program. Shouldn't program content also drive cycle times? It turns out that the answer is "yes"—but **the enormous increase in the complexity of our high-end systems has been offset by improvements in hardware design productivity**. Advances in computer-aided design and manufacturing (CAD/CAM), modeling and simulation methods, and physics-based design have all helped to dampen the cycle time implications of our appetite for complicated hardware.

Unfortunately, the same is not true for software.

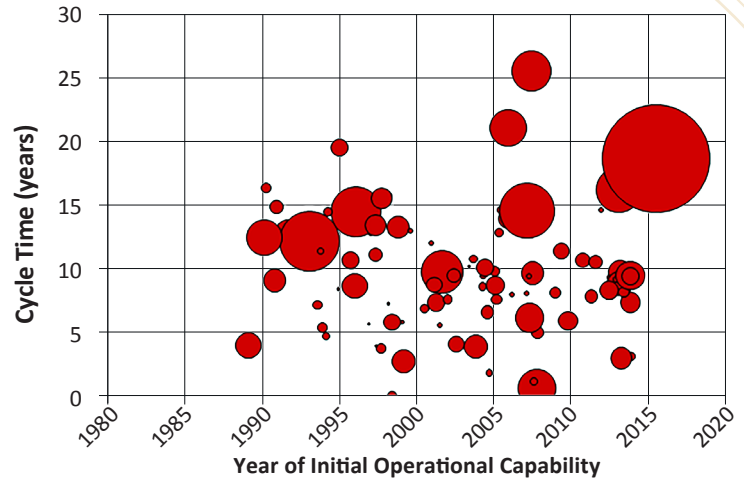
Over those same 30 years, the hourly productivity of software developers has not improved nearly as much as for hardware. Meanwhile, the exponential increases in available computing power have enticed us to shift more and more functionality from hardware to software. This was a fantastic trade when software was only a minor part of system development—it increased system flexibility and robustness while decreasing development cycle times.

But what happens when the software dominates the system?

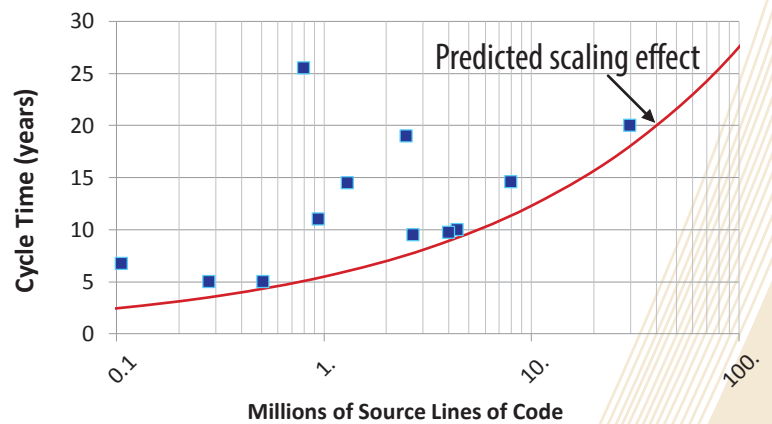
We have already seen the answer in our space systems. In the mid-2000s, the space system development community realized that space probes are now essentially software systems with hardware peripherals, and that software development and integration drive both cost and cycle time. **Attempts to reduce cycle time without reducing complexity generally have led to mission failure.**

If the same thing is not already happening for our major weapon systems, it will soon. Our best models of how long it takes to produce software of a given complexity show a sobering trend. **It doesn't take very many millions of lines of code before you can expect just the software to take a decade to be ready to field.**

So what can be done if prompt fielding is paramount? If software is now the time-limiting factor, we must either limit how much we ask for (i.e., by making explicit software vs. hardware trades early in design) or make major breakthroughs in software productivity. **Only changes to software—either in how much we ask for or how we develop it—will let us design and field complex systems more quickly.**



Cycle time for major defense programs sized by program cost



Software size & cycle time for recent programs

* Based on: IDA [NS D-5762](#) (Revised), "Acquisition Cycle Time: Defining the Problem," David Tate, October 2016. Research sponsored by the Office of the Under Secretary for Acquisition, Technology and Logistics (OUSD/AT&L).