Chapter 0. README

V1.4, 11 Jan 2019

Software is ubiquitous in the world around us and U.S. national security is critically dependent on the capabilities of its software. To maintain our military advantage, the Department of Defense (DoD) must be able to develop, procure, deploy, and continuously improve software faster than our adversaries. Recognizing that not all "software" is the same – it can range from off-the-shelf, non-customized applications to highly-specialized, embedded code running on custom hardware – it is critical that the right tools and methods be applied for each type. Commercial industry has demonstrated that software can have a transformative impact on business and society. Companies that thrive take advantage of software, computing, and networking – and the rapid cycles of improvement they allow – to the maximum extent possible. At the present time, DoD's software prioritization, planning, and acquisition processes are among the worst bottlenecks for deploying capability to the field at the speed of relevance. This puts the U.S. Armed Forces at risk, reduces the efficiency of DoD operations, and drives away the very people who are most needed to develop software that is critical to national security.

What this report is about. This manifesto describes the output of the Defense Innovation Board (DIB) Software Acquisition and Practices (SWAP) study. The DIB was charged by Congress[1] to recommend changes to statutes, regulations, processes, and culture to enable the better use of software in the DoD. We took an iterative approach, releasing a sequence of concept papers describing our preliminary observations and insights (the current versions of which are included in Appendix A) and using those to encourage dialogue with a wide variety of individuals and groups to gain insights into the current barriers to implementing modern software. This report attempts to capture key insights from these discussions in an easy-to-read format that highlights the elements that we think are critical for the Department's success.

This report is organized as follows:

- **TL;DR:** a one-page summary of 12 months of work for those not likely to read the full report; please take the time to read it. (TL;DR is Internet slang for "too long; didn't read").
- **README** (this document): a more detailed five-page summary of the report. If your boss read the TL;DR, thought it was intriguing, and asked you to read the entire report and provide a short summary, cut and paste this chapter and you should be good-to-go. A final bonus page has a list of the recommendations, so you can pretty much stop at that point – or better yet, stop after suggesting to your boss she adopt them all. (A README file is used by the open source software community to provide essential information about a software package.)
- **Chapters 1-4:** short descriptions of key areas we felt were important to expound upon. If you attach the TL;DR to any one of these as a preface, it should be comprehensible.
- **Chapter 5:** a more detailed description of the recommendations and our rationale.

[1] 2018 NDAA, Sec. 872. Defense Innovation Board analysis of software acquisition regulations.

- **Supplementary Information:** To ensure that the main body of the report satisfies the staple test[2] and the takeoff test,[3] we put most of the additional information generated during the study in a set of appendices. These provide a wealth of examples and evidence, but we took care to put our essential arguments up front for less wonky types.

Key Themes. In order for the report to be useful, we felt we should come up with a few key themes that could be used to drive home the message of the report. Here they are (again):

1. Software is ubiquitous and U.S. national security relies on software.
2. Speed and cycle time are the most effective metrics for software.
3. Software is made by people, for people, so digital talent matters.
4. Software is different than hardware (and not all software is the same).

Software is ubiquitous and U.S. national security relies on software. The rise of electronics, computing, and networking has forever transformed the way we live: software is a part of almost everything that we interact with in our daily lives, either directly through embedded computation in the objects around us or indirectly through the use of information technology through all stages of design, development, deployment, and operations. Our military advantage, coordination with allies and partners, operational security and many other aspects of the DoD are all contingent upon our software edge and the lack thereof presents serious consequences. Software drives the competitive advantage: what makes weapons systems sophisticated is the software, not (just) the hardware.

Commercial trends show what is possible with software, from the use of open source tools to agile development techniques to global-scale cloud computing. Our adversaries are active players in the world of software and so they are increasingly able to develop weapons systems faster than we can, capitalizing on their advantage in software development. Meanwhile, they exploit our vulnerabilities via cyber-attacks to steal, undermine, and inhibit our capabilities. The incoming generation of military and civilian personnel began life digitally plugged-in, with an innate reliance on software-based systems. They will demand new concepts of operations, tactics, and strategies to maintain the edge they need. If the Department can reform its acquisition processes and adjust its culture and personnel policies before its too late, this software-savvy generation can still set the Department on the right course.

Speed is the ultimate software metric. Being able to develop and deploy faster than our adversaries means that we can provide more advanced capabilities and be more responsive to our end users. Faster reduces risk by focusing on the critical functionality rather than over-specification and bloated requirements. It also means we can identify trouble earlier and take faster corrective action which reduces cost, time, and risk. Faster leads to increased reliability: the more quickly software/code is in the hands of users, the more quickly feedback can focus efforts to deploy greater capability, sooner. Faster gives us a tactical advantage on the battlefield because we can operate and respond inside our adversaries' observe–orient–decide–act (OODA) loops.

[2] Any report that is going to be read should be thin enough to be stapled with a regular office stapler.
[3] Reports should be short enough to read during takeoff, before the movies start and drinks are served.

Software is about people. As Steve Jobs observed,[4] one of the major differences between hardware and software is that for hardware the "dynamic range" (ratio between the best in class and average performance) is, at most, 2:1. But, the difference between the best software developer and an average software developer can be 50:1, or even 100:1, and putting great developers on a team with other great developers amplifies this effect. Today, in DoD and the industrial base that supports it, the people with the necessary skills exist, but instead of taking advantage of their skills we put them in environments where it is difficult for them to be effective. In DoD proper, we do not take advantage of already existing military and civilian personnel skill sets by offering pay bonuses, the ability to stay in their specialization, or access to early promotions. Skilled software engineers and the related specialities that are part of the overall software development team need to be treated like Special Forces and the United States must harness their talent for the great benefits that it can provide.

Software is different than hardware. Over the years, Congress and DoD have developed a sophisticated set of statues, regulations, and instructions that govern the development, procurement, and sustainment of defense systems. This process was developed in the context of the Cold War, where major powers developed aircraft carriers, nuclear weapons, fighter jets, and submarines that are extremely expensive and require tremendous access to capital and natural resources. Software, on the other hand, is something that can be mastered by a ragtag bunch of teenagers with very little money – and can be used to destabilize world powers. Currently most parts of DoD develop, procure and manage software like hardware, assuming that it is developed based on a fixed set of specifications, procured after it has been shown to comply with those specifications, and then "maintained" by block upgrades and new procure- ments. But software development is fundamentally different than hardware development, and software should be developed, deployed, and continuously improved using much different cycle times, support infrastructure, and maintenance strategies. Testing and validation of software is also much different than hardware, both in terms of the ability to automate but also in the potential vulnerabilities found in software that is not kept up to date. Software is never "done," and must be managed as an enduring capability that is treated differently than hardware.

The First Three Things to Do. The Department's current approach to software is a major driver, if not *the* major driver, of cost and schedule overruns for major defense acquisition programs (MDAPs). Congress and DoD need to come together to fix the acquisition system for software because it is the primary sources of its acquisition headaches.

Bringing about the type of change that is required to give DoD the software capabilities it needs to stay ahead is going to take a significant amount of work. While it is possible to use the current acquisition system and DoD process to develop, procure, deploy, and continuously improve DoD software, the statutes, regulations, processes, and culture are debilitating for software. The current approach to acquisition was defined in a different era, for different purposes, and only works for software projects through enormous effort and creativity. Congress, the Office of the Secretary of Defense, the Armed Services, defense contractors, and the myriad of government

[4] Steve Jobs - The Lost Interview, 2012.

and industry organizations involved in getting software out the door need to come together and make major changes. Here are the three most important things to start with:

1. **Create new statutes streamlined for software** that provide more insight while enabling rapid deployment and continuous improvement of software to the field (bear with us).

2. **Create cross-program/cross-service digital infrastructure** that enables rapid deployment, scaling, and optimization of software as an enduring capability, managed using modern development methods in place of existing (hardware-centric) regulations.

3. **Create new paths for digital talent (especially *internal* talent)** by establishing software development as a high visibility, high priority career track with specialized recruiting, promotion, organization, incentives, and salary.

None of these can be done by a single organization within the government, so they are going to require a bunch of hard-working, well-meaning people to work together to craft a set of statutes, regulations, processes, and (most importantly) a culture that recognizes the importance of software (theme 1), the need for speed and agility (theme 2), the critical role that smart people have to play in the process (theme 3), and the impact of inefficiencies of the current approach (theme 4). In many ways this mission is as challenging as any combat mission: while participant's lives may not be directly at risk in defining, implementing, and communicating the needed changes to policy and culture, the lives of those who defend our nation ultimately depend on the ability of the Department to redefine its approach to delivering combat-critical software to the field.

New statutes, streamlined for software. Congress has created lots of workarounds to allow the DoD to be agile in its development of new weapons systems, and the Department has used many of these to good effect. But the default statutes, regulations, and processes that are used for software too often rely on the traditional hardware mentality (repeat after me: software is different than hardware) and those practices do not take advantage of what is possible (or frankly necessary, given the threat environment) with modern software. We think that a combination of top-down and bottom-up pressure can break us out of the current state of affairs, and creating a new acquisition pathway that is tuned for software (of various types) could make a big difference. To this end, Congress and DoD should prototype and eventually create mechanisms for ideation, appropriation, and deployment of software-driven solutions that take advantage of the unique features of software (versus hardware) development (start small, iterate quickly, terminate early) and provide purpose-fit methods of oversight.

Cross-program/cross-service software digital infrastructure: We need to create, scale, and optimize an enterprise-level architecture and supporting infrastructure that enables creation and initial fielding of software within 6 months and continuous delivery of improvements on a 3 month cycle. This "digital infrastructure," common in commercial IT, is critical to enable rapid deployment at the speed (and scale) of relevance. In order to implement this recommendation, Congress and Department leadership must figure out some ways to incentivize the Services and defense contractors to build on a common set of tools (instead of inventing their own) *without* just requiring that everyone use one DoD-wide (or even service-wide) platform. Similarly, OSD is going to have to define some non-exceptions-based alternatives to (or at least pathways through) JCIDS,

PPB&E, and DFARS[5] that are optimized for software. DOT&E will need new methods for operational test and evaluation that match the software's speed of relevance, and CAPE is going to have to capture better data and leverage AI/ML as a tool for cost assessment and performance evaluation. Finally, the Services are going to need to identify, champion, and measure platform-based, software-intensive projects that increase software effectiveness, simplify interconnectivity among allies, and reform business practices. Subsequent chapters in our report provide specific recommendations on each of these areas.

New paths for digital talent. The biggest enabler for great software is providing great people with the means to contribute to the national security mission. While the previous recommendations speak to providing the tools and infrastructure DoD technologists need to succeed, it is equally important that the Department's human capital strategy allow them to even do this work consistently in the first place. Particularly important is to provide new career paths for digital talent and enable the infrastructure and environment required to allow them to succeed. The current GS system favors time-in-grade over talent, and this simply will not work for software. The military promotion system has the same problem. As with sports, medicine, and law, great teams make a huge difference in software and we need to make sure those teams have the tools they need to succeed and reward them appropriately -- through recognition, opportunities for impact, and pay. Advanced expertise in procurement, project management, evaluation and testing, and risk mitigation strategies will also be needed to create the types of elite teams that are needed. To get started, Congress should create a two-year national security waiver from the GS system in selected digital technology areas required for software and the Services should use this and other authorities to identify and nurture civilian and military talent with software development expertise. A key element of success is finding a way to keep talented people in their roles (rather than transferring them out because it is the end of their assignment), and promote people based on their abilities, not based on their years of service.

The Next Ten Things to Do. The items above are what we think Congress and the Department should focus on as the first three things to accomplish. Without dramatic change, the rate at which we can make improvements is far outpaced by the rate at which the problem itself gets worse. With demonstrated progress on these three there is then a long list of other things that need to be done, ranging from changing the law to changing the way people work. We created a list of 30 recommendations for change that we thought were important, and then asked everyone with whom we interacted in the building on this report to vote on the ones they thought would make the most difference. Here is the current snapshot of the top 10 recommendations based on that voting:

Rank	**Recommendation**	□	□
	This table will be filled in for the final report		
	The items here will come from a longer list of recommendations (see cheat sheet)		
	The order will be determined using a leaderboard, hosted on [TBD service]		

[5] Common DoD acronyms are defined in Appendix F (Acronyms, Inside Jokes, and Catch Phrases).

	Participants in SWAP study activities will be allowed to cast a vote		
	More details coming later; look at the full list of options (cheat sheet) for now.		

More details on these (as well as top 10 lists for the biggest barriers and the most useful tools that are not currently available for use) are included in Chapter 5 (What Would the DIB Do) and the supplementary information.

Getting Started Now. The types of changes that we are talking about will take years to bring to complete fruition. But it would be a mistake to spend two years figuring out what the answer should look like, spend another two years prototyping the solutions to make sure we are right, then spend two to four more years implementing the changes in statutes, regulations, processes, and culture that are actually required. Let's call that approach the "hardware" approach. Software is different than hardware and the approach to implementing change for software should be different as well.

Many of our DoD issues could be addressed by adopting existing best practices of industry for agile development, software as a service, use of modern (cloud) infrastructure, tools, computing and shared libraries, and software logistics and support delivery systems for software maintenance, development, and updating (patching). We do not need to study these, we need to get going and implement them. Here are some specific suggestions for what to do starting *now*:

- FY19 (create): High-level endorsement of report vision and support for activities that are consistent with the desired end state (i.e., DevSecOps and enterprise-level architecture and infrastructure). If you are reading this and are in a position of leadership in your organization, pass this on to others with your seal of approval and a request for your team to develop 2-3 plans of action for how it can be applied in your domain. If someone comes to you with a proposal that aligns with the objectives we have outlined here, find a way to say yes.
- FY20 (deploy): Initial deployment of authorities, budgets, and processes for SWAP reform. Choose immediate representative programs to act according to the themes, flavors, and recommendations in this report, implement now, measure results, and modify approaches. Let's implement this report the way we implement modern software.
- FY21 (scale): Streamlined authorities, budgets, and processes enabling SWAP reform at scale. In this time frame, we need a new methodology to estimate as well as determine the value of software capability delivered (something not based on lines of code).
- FY22 (optimize): All DoD software development projects transition (by choice) to software-enabled processes, with talent & ecosystem in place for effective management & oversight.