How the Pentagon could think about Software Development

Deputy Secretary of Defense Dr. Hicks spoke in her announcement of the DOD AI and Data Acceleration initiative (ADA) that, "A key part of an AI-ready department is a strong data foundation." In the 2020 DOD Data Strategy, the department presented the DOD's data goals through VAULTS (Visible, Accessible, Understandable, Linked, Trustworthy, Interoperable, and Secure). The key to making both of these happen is software. In this context, software is the entire range from extensive COTS tools, like Microsoft Excel, all the way to a few lines of python code an analyst writes to create a report. Currently, the Pentagon treats both ends of the Software spectrum very differently. In simple terms, data is visible, accessible, understandable, linked, trustworthy, interoperable, and secure by software regardless of the scale. The Pentagon needs to make software production as easy as writing a memo. Risk is inevitable; however, hiding behind the excuses of security is an even greater risk.

1. Make all software development tools employees want available
2. Empower people across any job role to build small applications with accessible interfaces
3. Provide ways to release software on all networks quickly

These ideas are not new or innovative; most of these are already implemented across the department when you go to specific agencies or commands. Again, this shows that we have the expertise to do this; however, we need to empower civilian employees and service members across the entire Department.

Make all software development tools employees want available

The key term here is "want." Working as a software developer in any other context allows you to set up your preferred working environment. The department has several service providers that have their lists of approved software, which always seem to lack the tools you are used to working with. If you get lucky, the service provider can suggest an alternative on the approved list, a version a few years out of date that may only match part of your requirements.

On the other hand, you may run into a scenario where your specific job role is unauthorized for using this software, or worse, it does not exist. There is currently no incentive for the service provider to improve this process.

On the odd chance that the requested software is approved, most service providers have to install it on your specific computer, where then you have to cross your fingers that the person assigned to your ticket knows what they are doing.

A solution that has proven to work is to host private copies of software on all of the distinct networks and mandate that all service providers allow self-installation from this hub. In addition, service providers need to reduce current restrictions on command line use, programming languages, and automated access to file systems, among others, to best facilitate this.

This could look something like this:

- tools.dev.mil - Entire executables;
- pypi.dev.mil - Registry for Python packages;
- npm.dev.mil - Registry for Node packages;
- docker.dev.mil - Docker Image Repository;

These are examples of domains on NIPR and should be mirrored on other networks simultaneously. A solution like this is possible because of the wide use of open-source software. Changing the language from "need" to "want" and focusing on the "want" will slowly build an entire infrastructure of valuable tools that will accelerate the production of in-house software.

Empower people across any job role to build small applications with accessible interfaces

Analytics are not static products; they are dynamic and should be treated that way. Freezing analytics by writing reports and printing numbers is inevitable; however, the data and analytics do not have to be. Creating a data-centric department presents the opportunity for all employees to learn how to automate portions of their workflow. This can be done through click and drag tools or a scripting language such as python.

Not every analyst needs or will want to write code. Still, every analyst should have access to a community of development to learn or work with developers to produce accessible products. This could look like a weekly meet up where analysts could sit side by side with a developer and build a quick prototype of an analytic or automation, or at the same time learn from a community presenter on how to make a Jupyter Notebook to rerun analytics needed for their day to day.

Accessing databases from any domain has been an enormous problem in the department, mainly stemming from legacy systems that were never designed to be used by external software. By empowering people to build small accessible applications, the Department can get ahead of this problem. Starting with the premise that every application is an Application Programming Interface (API) before a web-based application removes this lockdown. This, however, introduces a new problem in the security space. The big concern is having individuals access data they should not have or tracking what information was extracted from systems. Expecting each developer or analyst to solve this problem themselves is a recipe for disaster and data leaks.

In the same vein as self-hosting outside packages and software, the department should develop DoD-specific modules and libraries for situations like this. A standard library for Common Access Card (CAC) certificate authentication and hosting authentication services to provide short-term certificates that individuals can apply for works hand in hand with giving developers the tools they need to succeed while keeping good security controls. Such services and libraries should be developed centrally to ensure the security standards are kept up to date.

Building a community of sharing and educating could look something like this:

- dev.mil - Hub for all DoD developers;
- stackoverflow.dev.mil - Knowledge sharing space;
- gitlab.dev.mil - DoD specific code sharing; etc.

Recommendation

Replace with "20%"

I am looking specifically at code sharing and the use of open-source software. The department has a congressional mandate to open source 25% of code developed internally. Therefore, when feasible, public places like GitHub and GitLab should be utilized to make code accessible even outside of the DoD. Using open-source technology comes with the understanding of giving back, which the department has failed at so far. The Defense Digital Service has already outlined an easy path to do so (https://code.mil/how-to-open-source.html). The department has even started this on GitHub; it lacks a more considerable commitment across the board (https://github.com/deptofdefense).

Provide ways to release software on all networks quickly

The final step of any piece of software is the release. On the open internet, this is as simple as acquiring a domain name from a registrar and hosting the application on something like a virtual machine of a cloud provider. However, the DoD's process involves an undetermined amount of steps and approvals. The process is also seemingly designed for significant pieces of software built by huge teams. It should not take lawyers, and security experts to release a web application that can be developed in a matter of hours.

It seems feasible that every developer should have access to a personal domain to deploy an application of an overview of what they work on and then have the ability to deploy applications on subdomains from there. We can look at this as a testing ground for applications or, as the department likes to put it, a "non-authoritative" source. It should then be easy to apply for a permanent domain name to bring the application into an "authoritative" state.

Having a domain to deploy to isn't enough, though. Each developer needs access to virtual computing stations to deploy these applications, which must be accessible on the respective network it's deployed on.

- myportfolio.dev.mil - Developer's personal homepage
- sampleproject.myportfolio.dev.mil - Project in development that anybody on the network can access
- finalproject.dev.mil - The production version of a product

and so on …

Next Steps

The department should have an entire office dedicated to making software available on all networks, The Office of Software Enablers. This office should have three focus areas: tools, people, accessibility.

There should be no distinction between job titles for employees that want to partake in software development. The goals of this office are measured in each area by:

- Tools
    - Number of requests approved and software published
    - The difference in time new versions are available to be published
    - Accessibility to the tools across the department
- People
    - Growth of the community developers
    - Software Development Hires
    - Number of projects developed for Open Source
- Accessibility
    - Number of sample projects converted to full production
    - Time from development to published on respective networks
    - Time of acquiring domain names

The department must take significant steps to have a strong software development focus, which sets the foundation for a data-centric department. A focus on these three areas will kickstart a movement to put the department on track to attract talented developers that can have a tangible impact on the operations across the entire Department of Defense.