Analytic Capabilities for Improved Software Program Management

Planning and Management by the Numbers:
Incorporating New Models of Risk and Causality into DoD Acquisition

The Problem

The ever-expanding reliance on software to achieve critical system capabilities is driving the necessity for software engineering organizations to adopt more robust analytic techniques to plan and manage their systems and software deliverables. Department of Defense (DoD) system and software acquisition and development are undergoing a transformation. The Software Acquisition Pathway in the DoD Adaptive Acquisition Framework (AAF) (Figure 1) illustrates how sprints and releases create a rolling wave of software that flows into the operational baseline. This relatively new approach to the development of software for DoD systems typically uses a DevSecOps pipeline.

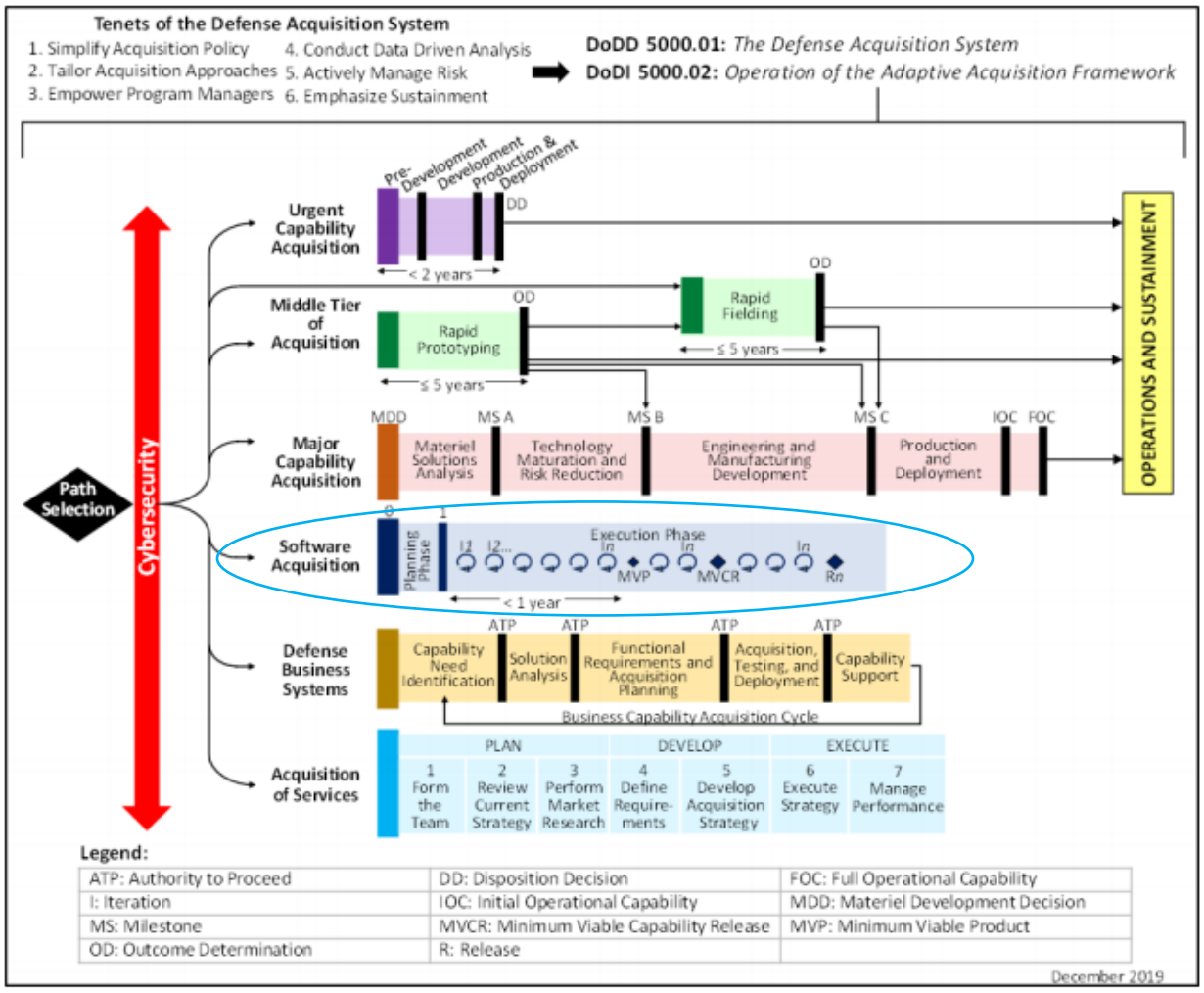

Figure 1: Adaptive Acquisition Framework

As a result of this transformation and innovation, there is little history and experience within the DoD software acquisition community to draw on to establish a quantitative foundation for planning and management of systems and software using the alternative approaches provided by the AAF. This dearth of data and experience negatively impacts cost estimation, but cost estimates per the DoDI 5000.73 are still required under the new software acquisition policy (DoDI 5000.87). Nonetheless, there is great potential for automatically collecting data from DevSecOps pipelines and martialing it for the purposes of estimation, planning, and management. The key will be collecting the right data and analyzing it in the right way. But collecting the right data is only one part of the challenge. New analytical approaches and models are also needed for estimating, planning, and management.

As software engineering practices and technology move towards more frequent release tempos, management also must keep pace and have the capability for timely, efficient, and impactful forms of program analysis and assessment. Software engineering organizations will require highly automated and instrumented engineering environments that provide the data needed for planning and management. While today's DevSecOps pipelines produce a lot of data, it is rarely used for planning and management purposes.

Having the right data does not automatically translate to having actionable information. Much of today's guidance on planning and controlling software projects is from programs following a more traditional hardware-centric acquisition lifecycle and is based on heuristics derived from correlation-based analyses, anecdotes, and experiences. As the dynamics within software projects change, so too must the guidance on planning and control. Relying on guidance grounded on *correlations* makes it uncertain that mitigations will impact the actual *causes* of problems such as cost overruns, schedule delays, and technical performance shortfalls. New methods and models for estimating, planning, and managing programs are needed that meet the demands of programs using the AAF.

A Novel Approach to Software Project Estimation and Management: Probabilistic Causal Models

Quantifying Uncertainty in Early Lifecycle Cost Estimation (QUELCE) is a Software Engineering Institute (SEI)-developed cost estimation approach based on modeling projects costs as a Bayesian Belief Network (BBN). This approach has a number of virtues but chief among them are (1) incorporating uncertainty into the inputs and assumptions associated with factors affecting project execution and success and (2) utilizing Monte Carlo methods to create an empirically derived distribution of predicted cost. Furthermore, BBNs can draw on subject matter expert judgements when data are scant or missing. More recently, the SEI has been conducting research using a form of AI called causal learning (CL). CL methods discover likely causal relationships among observational (vs. experimental) data such as those in a project database or generated from a DevSecOps pipeline. Similar to a BBN, CL methods produce graphs. When used to analyze project data the graphs represent the causal relationships and dynamics as determined from the project data. Causal learning can help organizations move from correlation-based models that are often *presumed* to reflect causal relationships to empirically grounded causal models for guiding action and making decisions.

Combining these two technologies, BBNs and CL, provides a unique approach to cost estimation based on probabilistic causal modeling that incorporates uncertainty and causality. Also, the method yields an

executable model that can be used to explore and evaluate scenarios of acquisition strategy as well as risk mitigations and corrective actions during development.

Supporting the method is a growing body of research results that identify and validate the factors and relationships that are vital to planning and managing software projects. Focusing on these factors should increase the effectiveness and confidence in cost estimates, plans, and management that utilize this modeling approach. It is also shedding light on heuristics and beliefs that are not supported by actual project data. In cases where unvalidated assumptions are, in practice, false, cost estimates are likely to be further off the mark and corrective actions taken by program managers are likely to be insufficient or even counterproductive. Understanding and acting causal relationships provides a more solid foundation for decision making.

Summary

The ever-increasing role of software in DoD systems is driving the necessity for software engineering organizations to adopt more robust analytic techniques to plan and manage software deliverables, quality, and overall software engineering capabilities. Today's drive towards the Software Acquisition Pathways and DevSecOps makes these models highly valuable and relevant. BBNs are well established and have had a large impact in a wide variety of fields. They have the virtue of being able to accommodate subject matter expert judgements. Causal learning is relatively new but has already affected cancer research, genomics, and economics; recent improvements in the causal learning algorithms now bring much of that same power to software engineering at a time when quantitative advancements in project estimation, planning, and management are needed. But, modeling and analysis require data.

Increasing automation and integration of tools is the wave of the future. The ability to harvest and analyze data from the tool chain is a relatively untapped opportunity with tremendous potential to yield detailed insight into software development programs. But not all influential factors and drivers of performance are captured within the tool chain. Some reliance on external sources and subject matter expert judgement will remain. Over time, with the right data collection, the reliance on judgement may be reduced. The combination of BBNs and causal learning together provides a practical way to learn from (1) all of the data generated by modern engineering environments and (2) to complement that with other sources as necessary. This yields a robust analytic capability that with high quality and high confidence planning and management guidance for software development projects.

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

DM20-0992