SECURING EVERYTHING AS A SERVICE

The Software, the Process, and FISMA Compliance

In the world of XaaS, *the internet of things*, and particularly in light of recent developments in the cyber-security language of contracts I've recently reviewed for capture; I've spent some time revisiting the literature and my thoughts surrounding this brief essay I'd written in early 2012 as an argument against the application of negligent entrustment in outsourc(ed|ing) IT.

Due, in part, to the presedential cybersecurity directives and appearing to be in response to changes in the National Defense Authorization Act, anticipated changes to the DFARS (originating with 2011-D039), and the continued focus on cyber security within all national sectors; a renewed focus on the development lifecycles and standards of programs warrants review and increased level-of-attention that extends beyond QA and EVM.

From a broader perspective, it becomes necessary to fully explore secure development lifecycles (SDL), the role of change management (CM) and the application of supporting standards, frameworks, or models (e.g. ANSI-748 EVM, Agile EVM, Scrum, CMMI, Microsoft's SDL, etc.) in governing program execution, while facilitating high-performing teams.

BACKGROUND

At issue, and the bits that got me working back into this subject, are the concepts of warranty when included in contract language and the very real potential of severe ramifications for contractors and consultants alike. Should they be found failing in application of governance and policy the penalty can vary anywhere from a poor contract rating, which impacts future capture, to payment refunds, or the ultimate death penalty of disbarment from award eligibility.

For reference, an example from the US Transportation Command is provided as follows:

```
The contractor represents and warrants that the software shall be free from all computer viruses,
worms, time-outs, time bombs, back doors, disabling devices and other harmful or malicious code
intended to or which may damage, disrupt, inconvenience or permit access to the software user's or
another's software, hardware, networks, data or information.
```

For the purposes of this short paper; the goal is to ***begin a discussion*** on frameworks supporting effective, secure, and practical development of software that meet all necessary areas of compliance. Under primary consideration are regulatory, contractual, and corporate governance; with methods (either technological or procedural) of audit, verification, and review that can (or should) be included in the process to ease implementation through incorporation into a standard workflow. The intent is to share my thoughts on integration and the effectiveness of aligning the multitude of control requirements in ways that don't negatively impact project execution and velocity.

DISCUSSION

While seemingly straightforward, there are second and third-order considerations that are often overlooked during capture, evaluation, and execution of programs that should require any of us in business, IA, software development, or any other affected field to take pause.

Take, for example, a malicious employee who fears termination. It's possible (and some might argue likely) that they slip a backdoor into a computer program. While this certainly represents a criminal act;

an entity that fails in applying their own policies or processes, or simply fails to implement effective controls can be held liable - and at great cost. Similarly, malicious injection on the part of a third party would most-likely be a criminal act; failure to identify the injected source, and to ensure that delivered application code is "free from all ..." remains the duty of the contracted party.

Given the increased attention on industry standards (e.g. ISO 27001, ANSI-748, etc.), cyber-security, and other domains of interest; issues of liability arise related to what may initially seem to be otherwise unrelated. CMMI, for instance wouldn't appear to be a protection against disbarment any more than effective Agile development supports EVM. Viewed holistically, however, each builds upon the other to develop a new kind of "defense in depth" where the defense isn't strictly within the realm of cyber, but extends into effective quantitative management of programs, and reasonable review of data to ensure a firm is practicing all necessary due diligence and control of program efforts.

These issues could be considered secondary to those that can be accommodated by best-practices in development and quality assurance (QA) and specified through frameworks and maturity models such as CMMI. When considering the following, however, they necessarily should be included.

> If the Government determines, after a security audit (e.g. ST&E), that software delivered under this task order is non-secure, the Government will provide written notice to the contractor of each non-conformity.
>
> Software shall be "non-secure" under this task order if it contains a programming error listed on the current approved version of the CWE/SANS TOP 25 (which can be located at http ://www.sans.org/top25-programming-errors) or a web application security flaw listed on the current approved version of the OW ASP Top Ten (which can be located at http://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project).

Given that traceable changes to source (SCM), peer review, and clear documentation are among the most effective methods to validate requirements, prevent malicious injection, and to reconcile findings from static analysis; it's worth noting that these are key concepts within most software QA and CPI frameworks. This traceability additionally supports progress-to-plan reporting and quantitative program management. At the end of the day then, integrating and orchestrating existing best practices from the major domains of governance, serve the goals of information security in this context.

To achieve convergence, information security professionals need a new way of thinking and supporting frameworks and tools that describe a greater role of governance applications. Akin to the growth of the business analyst's role, often including systems, requirements, and process engineering subdomains, IA professionals must remain aware of the broader scope of technological and procedural avenues to achieving compliance. We must additionally understand the industry, applicable regulatory and legal issues, and the goals of the enterprise well enough to support efforts in process re-engineering in ways that can deliver value.

A New Market Advantage?

While this type of compliance, if enforced, could certainly be considered a program risk; I'd prefer to think of it as a strategic opportunity for those that lean forward in developing their capabilities in software and systems audit. While all companies necessarily innovate, what I suggest is more transformative in nature.

The National Cybersecurity and Critical Infrastructure Protection Act of 2013 contains guidance, and further opens transparent dialogue in public-private information sharing in other sectors. Consider the ramifications should information security and assurance extend beyond the necessary requirements to meet existing regulatory requirements such as SOX, GLBA, etc., and beyond industry's self-regulation in the form of PCI or BITS SAP. Given the criticality of IT to most organizations, the basis of software behind most of these systems, and the emergent change in requiring some degree of accountability from the developers of these systems; it's nearly a foregone conclusion that additional guidance will be developed, that reporting and audit will be required, and that positive control/traceability of control effectiveness will be necessary to protect firms from liability.

If we, as leaders, consider these scenarios to be reasonably likely; it stands to reason that we should be planning for disruptive changes in the development and delivery of solutions to our customers, and ensure appropriate control structures are developed and enforced internally. For those of us that provide: business services, consulting, software engineering & integration, or any number of other services, it additionally stands to reason that the adoption of orchestrated and auditable processes, enabled by technological integration, reduces the cost of compliance; using the overhead to create value. Additionally, the systematic application and quantitative management of these (as with any control framework) encourages continuous improvement to develop of capacity, opening the doors for the creation of a new set of offerings based on internal capabilities.

When disruptions within an industry is seen as opportunity for transformative change, rather than a threat to the bottom line; there is certainly rationale to begin development of new offerings as early as possible, and to become involved in shaping the final outcome of required compliance.

Particularly, situations such as this where there is potential that a large, sustained, and **relatively unfilled** market will soon exist; created through legislation akin to the financial and health auditors who validate compliance with SOX, GLBA, HIPPA, etc., it stands to reason that those firms adopting and developing these capabilities now will have a significant advantage.

Final Remarks

From a DoD contractor's perspective it's important to note that these issues are particularly problematic in low-price awards or variations where evaluators don't have the flexibility to consider them within the evaluation of staffing, technical approach, or price realism assessments.

Given the fact that the majority of RFPs do not specifically include this type of language, yet the requirements still exist under FISMA, and by the inclusion of the FAR clause(s) requiring compliance with DoD 8500.2 by reference (or referenced reference) it is difficult to justify the resources needed to comply with all controls within a cost narrative or delivered Basis of Estimate (BoE). Until mandatory compliance is set as an evaluated criteria in all contracts, rather than being included strictly within the statement of work there remains limited opportunity for execution of such a plan. Similarly, until we, as contractors highlight this oversight (or until more severe breaches occur) – a change to include this language is similarly unlikely. It still remains a valuable, and important, area of change in organizational process; with opportunities likely in various markets.

I'll certainly revisit this subject in the near future as I develop the actual security plans, compliance matrices, overlap charts, and guides for framework integration. In order for potential solutions to be applicable and available, my intent is to use open-source-ALM tools to orchestrate processes for development, tied in to ECM, BPM, and BI utilities for routing, reporting, knowledge management, and analysis. Look for a review of the available solutions in each of these spaces as the next parts in this series.

//Levii

Permalink: http://levii.com/269/negligent-entrustment-revisted-thoughts-for-software-development-contractors