Becoming a Smart Buyer of Software
Transcript

Part 1: Buying vs. Building Software

Julia Allen: Welcome to CERT's Podcast Series: Security for Business Leaders. The CERT Program is part of the Software Engineering Institute, a federally funded research and development center at Carnegie Mellon University in Pittsburgh, Pennsylvania. You can find out more about us at cert.org.

Show notes for today's conversation are available at the podcast website.

My name is Julia Allen. I'm a senior researcher at CERT, working on security governance and executive outreach. Today I'm pleased to introduce Brian Gallagher, director of the Acquisition Support Program at the SEI. We'll be discussing what business leaders need to know when acquiring or purchasing software instead of developing it within their organizations, including implications for security. So welcome Brian, glad to have you with us today.

Brian Gallagher: Well, thank you Julia. I appreciate the opportunity to do this.

Julia Allen: So I'll start off here. In today's business climate many organizations are purchasing software obviously, or software-intensive systems and software services, instead of growing their own. They're just finding it to be more cost effective. So in your experiences what are some of the key differences in these two perspectives, buying versus building?

Brian Gallagher: I think you hit the nail on the head right there Julia, when you said there are some key advantages, economic advantages really are what you're looking at. If you're an organization that has a business or mission set that doesn't necessarily align with developing products or IT systems, it doesn't make a whole lot of sense to apply a lot of management attention to having a development organization in-house.

So there are advantages to going outside to experts who really understand system development, IT development, providing IT or system's kind of services, rather than doing those in-house. And the Department of Defense especially has been doing this for a number of years. Instead of having a standing army of developers, they'll go to experts who have specialties in certain areas and take advantage of that. So there are some business advantages to doing that.

And when you think about the similarities and differences, I think there's a lot more similarities with what you have to do as an acquirer versus what a developer does, and those similarities really have to do with the engineering activities and the management activities. You can't just turn the keys over to a developer and say "Bring me something back in five years that'll meet my business objectives." You really have to be proactive about understanding the needs of the business, translating those needs and threats and new opportunities into requirements that somebody can go build a system to, providing leadership during the life of that program, and then transitioning that system back into the business.

So the kinds of things you need to be aware of as an acquirer are systems engineering activities, architecting activities, program leadership, transition type activities, and then maintenance and support type activities. So it's not that you now no longer have responsibility – you still have a major responsibility to ensure success.

Julia Allen: Well, that makes a lot of sense, because I think sometimes people, organizations who are buying software believe that they can really outsource everything, including all the responsibility, all of the knowledge. And what I hear you saying is you really need to have core competencies or skills in some of the same areas if you're buying versus building, correct?

Brian Gallagher: That's right, and I think that's a major mistake that people make, is thinking that they no longer have to have any kind of capability in that area. And you can think of it as kind of a lean or an agile management layer that you have to retain within your organization. Because you really know what the business needs are, the operational threats, and new systems and capabilities ought to address those needs and threats. And you can't just expect that somebody else with different business objectives will be able to come in and understand what those are in your context. So that up front work, the leadership that you provide during the life of the program – because your needs evolve and you want to make sure that those needs are expressed throughout the life of the program and are being addressed by the developers.

And then the major thing that we mess up on is transition activities. Many programs find that when they show up with the product, just trying to get that installed or get it transitioned into the business of the organization, is a major accomplishment that's underestimated and underappreciated.

Julia Allen: Well, you started down this path of the acquisition lifecycle, so let's keep going with that. So, clearly the purchasing or the acquisition is really only the first step in the process, and as you started to describe, I need to be able to effectively manage this relationship for as long as it exists. So can you say a little bit more of [what are] some of the key challenges and issues across the acquisition lifecycle?

Brian Gallagher: Sure, and first let me start by kind of explaining what we mean by acquisition. And we take the same view that the Department of Defense uses for acquisition, which is really when you think about it from the "I need," or I want," to the "I got." So that's a pretty wide range there. It's not about contracting or purchasing. It's really about being able to translate the needs of the organization into something that somebody can build to, getting that on contract which is a small part of that (which some people think is the major part of acquisition).

But then after that relationship is established, there's a win-win relationship that needs to continue through the life of the program, that leadership – the technical leadership, the business savvy – what is it that our mission is and how do we translate that to the other organization that's doing the development through the life of the program. And then that acceptance piece, and making sure that the change management activities that are happening within the organization as they adopt this new technology. So it's really from cradle to grave, if you think about it from that perspective, and in the DoD they even extend it beyond that into disposal of the system. So from the "I want" to the "I got" to even the life of disposal.

So when you think about it from those perspectives, it's lots of challenges and lots of opportunities for I guess mistakes along the way, just trying to translate some kind of business need into requirements is a very tough job. And a lot of organizations will fail at that very early part to capture what is the real operational need or threat that we're trying to address. Whenever you translate that into a set of requirements that you then have to hand off to somebody else, there's room for interpretation and sometimes that interpretation means that the contractor will make assumptions that are wrong. So there's lots of room for error in that process, from need to requirements to contractual requirements that sometimes don't speak well to the need.

Julia Allen: So it sounds like you really have to put the same type of program management, processes, practices, oversight, reporting mechanisms in place for this type of relationship where

someone else is building a piece of your system for you, as you would if you were doing it in-house, maybe even more so?

Brian Gallagher: I think that's right. And it's also kind of higher-end type activities, so if you look at the kinds of skills that you would need to do those activities, this is not just somebody who is just out of college. You really need somebody who understands systems engineering and architecting, can understand what the real business needs are and translate those and make sure that the systems being developed meet not only the technical needs, if you've got an existing architecture, but also the business needs of the organization. So it really takes some higher-end kind of thinking that you need to retain in-house.

Julia Allen: Well, and I would also expect that you need some fairly adept people skills because sometimes managing an outside relationship with another organization can be much more challenging than managing folks within your own shop.

Brian Gallagher: Absolutely, and in many cases you've got multiple contractors, or suppliers, or vendors – however you describe them – that have to work together and you're orchestrating all that. So not only do you have that relationship with the supplier, but you've also got to kind of facilitate the relationship between multiple suppliers, and then the operational aspects of the organization that you're representing. So, it is kind of a people skill thing as well, so not only the high-end management and engineering, but also the people skills and the organizational change I think is something we don't address well enough in this area also, is you really have to have some organizational change skills.

Part 2: Acquiring Software with Security in Mind

Julia Allen: Well that's a great overview and introduction Brian, to some of the issues, concerns and a bit of the upside and downside. So given that this is a security podcast series, why don't we talk a little bit more specifically about security. So, in your view what should a business leader think about, or perhaps what key questions should they ask, if security is an important requirement for the software they're intending to acquire or purchase?

Brian Gallagher: Well, I think the first thing they have to do is start asking some internal questions, and that has to do with what the operational need is. So if you've got a system that has to operate in a more secure kind of environment, whether it be the physical security or the data security or some other aspect that you're concerned about, you really need to understand what that is. Obviously the most secure system is one that nobody can use. So there's some tradeoffs you have to make, really understanding what the environment is that you're trying to get this new system to work in. And I think that's one of the real key challenges, is trying to understand the operational need and perspective in that manner.

There's also some things you need to think about, the questions you need to ask your vendors. You're not just concerned, I don't think, about the product and the security attributes of the product, but also the developer's ability to build high quality software that's free of vulnerabilities.

Now if you're just buying something off the shelf I think there's a different set of questions you have to ask about the pedigree of the software. Are there second and third tier vendors that provide products to this? And how do you know you've got all the licensing correct? And how do you know that the system functions as it's intended? But if you're developing a new capability, you want to make sure that the developer has the skills to understand your operational concerns and the environment that you're working in from a security perspective, but also the quality of the system. And I think one of the things we've found are many of the vulnerabilities that we have in systems

today are just due to sloppy development practices. So, thinking about quality assurance, systems assurance, software assurance kinds of activities during the development process and does this vendor that I've selected have the capability to provide those kinds of skills, is an important question to ask, really before you even award the contract.

So if that's important to you, you've got to make that part of the selection decision as you're looking for a developer who can provide that capability. Have they done this before? Can they give you examples of where they've developed high quality, high assurance kinds of systems? And do they have the ability to understand what your operational needs are with respect to security?

Julia Allen: Well, that's a real nice build to my next question, which is have you seen particular oversight processes or contractual mechanisms or other approaches that can help an organization that wants to acquire secure software?

Brian Gallagher: Well, I think as I said, the first thing you need to do is understand what the real operational need is – so techniques that help you understand the operational concept, the threats to the mission, and being able to translate those kinds of things into contractual requirements. There's a couple of things I know within the CERT Program, you've got some techniques for doing operational threat analysis. We also use something we called a Quality Attribute Workshop. The quality attributes are those things that are related to the product that are not functional characteristics.

And I think all too often we focus on the functional behavior that we expect, and we don't think about those other things, the quality attributes - some people call them the – ilities – that make the product.

Julia Allen: You mean like performance and reliability and dependability - things like that?

Brian Gallagher: Exactly, so if you've got a system, let's say, that's going to be used by a soldier out in the field, maybe a handheld system, what are the quality attributes that you expect that system to display? Not just the functional behavior, but let's say the performance, let's say it's a situational awareness system, and you want to make sure that the data is fresh, and what do you mean by fresh? So really exploring scenarios and use cases that drive those quality attributes out early on. So when you're thinking about the operational needs you've got to think about from an operational perspective the scenarios and use cases that are non-functional related.

So if you could imagine this handheld system that a soldier might have – how are you going to maintain that? How are you going to provide updates to it? – those kinds of things are non-functional requirements. The other thing to think about – what if it gets in the hand of the bad guys? So now you've got a system that tells you where all the friendly forces are in the hands of the enemy. And so you need to think about from that perspective how do you protect that information from being used against you in that kind of environment.

Julia Allen: So if I had those types of requirements – I really appreciate your example – would those show up in, for example, my requirements specification? Is there something like a service level Agreement, in design documents that I as the buyer or acquirer of the software would use that? How would I capture those kinds of specifications?

Brian Gallagher: Well, there's a couple of ways to do that. Typically we'll see, along with the statement of work that kind of describes what the system is going to be and what the expectations are for delivery and interfaces and so forth, you'll also usually see some kind of specification – a functional requirements spec, or a technical requirements document – that's provided as part of

the statement of work. And within that requirement spec or that requirements document, you'll have all those functional requirements we talked about.

But you also want to make sure that you include in there a description of these quality attributes. And it's not just "the system must be secure," or "the system must interoperate," or it must have this level of performance. But it's also got to be under what conditions, and who do we need to interoperate with, and how secure does it need to be. So it's more than just a statement of the requirement. It's as well a scenario that describes the use of that and what the expected outcome is in that environment.

Julia Allen: And then as the relationship progresses, what types of reviews or measures or reporting mechanisms have you seen be successful for ensuring that those requirements are being fulfilled?

Brian Gallagher: Well, I think there's two things you need to do, and we touched on this a little bit earlier about kind of making sure that the developer has the experience to, demonstrated experience to develop a system of this type. During the life of the program now you've selected a vendor, you're working with them. You've got to be concerned both about the product that's being developed, and so you want to do product reviews, technical reviews of the product to make sure it meets those specifications.

But you're also concerned about the process that the developer is using to make sure that they're using good coding standards, good coding practices, that they have a really world-class quality assurance kind of activity to make sure that the system is free of defects and vulnerabilities. And that defect management piece is one that we tend to overlook, really understanding the defect profile, the way the system will mature during the life of the program, and understanding that reliability growth through the life of the program is key to understanding how, what level of quality you have. So what you need to do is focus both on product and process.

The other thing you need to think about, and this is something that we tend not to do very well, is not just think about how the developer process is working and how the product is coming out, but also you need to think about what vulnerabilities do you introduce as the acquirer to the process. So during the life of the program there may be some design artifacts per se, or some other activities that you have access to that if, let's say in the case of the handheld system out in the battlefield, if the bad guys get access to the designs because you're not treating that as an assured part of your process, that could create problems operationally. So there's aspects you need to think about from the acquirer perspective. How do I protect the information? How do I protect the designs? How do I make sure that my process doesn't lead to vulnerabilities down the road?

Julia Allen: Well, that's a great point, because we tend to focus on our vendor or our supplier or our contractor, and think that we either have our act together or don't think about the risks that we ourselves can introduce as the acquirer.

Brian Gallagher: That's right. The Achilles heel to many of these systems are those vulnerabilities, and I know you guys have worked quite extensively on that. But exposing those vulnerabilities is a danger.

Julia Allen: Well, Brian, this has been great; just one last question to bring our conversation to a close. How can our listeners, in your opinion, or based on the resources that you see, become better-informed buyers of secure software, secure systems and services? Where can they learn more about this subject?

Brian Gallagher: Well, there's a couple of places. You mentioned cert.org, I think earlier on, which is a great resource for if you're concerned about for example secure coding practices, if you're a developer, you can go there and understand what that means. If you're concerned about some of the things you need to do as an acquirer to lower kind of the insider threat, there's a whole series of activities on insider threat that I think are key.

One key resource is the Department of Homeland Security's Build Security In website. I've looked at that many times, and there's also an area in there on acquisition, it's a content area. Some of the folks that work in my program have contributed to that. Rita Creel is one of our chief engineers who works in the intel area. And she's written several articles, one of them which kind of addresses one of the earlier questions you had, is about security issues and considerations in the lifecycle. So it goes through the entire acquisition lifecycle, talks about the up front early activities, and what are the security considerations there, what do you have to think about during the life of the program, some of the transition issues.

The other thing you might want to look at is at the SEI website, www.sei.cmu.edu. There's this whole section on acquisition, not specifically to security, but there are some practices on the quality attribute determination. There's a survival skills course that we offer for acquirers and there's lots of resources there that you can look at.

Julia Allen: Well Brian, this has been terrific. You've sure given us a lot to think about, and I appreciate very, very much your time and expertise, and I look forward to talking again.

Brian Gallagher: Well, thank you very much.