



## Acquisition Archetypes

featuring Bill Novak interviewed by Suzanne Miller

---

**Suzanne Miller:** Welcome to the SEI podcast series, a production of the Carnegie Mellon Software Engineering Institute. The SEI is the only software-specific, federally funded research and development center and is located at Carnegie Mellon University in Pittsburgh, Pennsylvania. A transcript of today's podcast is posted on the SEI website at [sei.cmu.edu/podcasts](http://sei.cmu.edu/podcasts).

My name is [Suzanne Miller](#). I am a principal researcher here at the SEI, and today I am pleased to introduce you to [Bill Novak](#), a senior member of the technical staff in our Software Solutions Division. Bill supports [acquisition-program](#) clients [by] assessing programs, teaching classes, and conducting research and development on the dynamics of acquisition-program behavior. In today's podcast, we'll be talking with Bill about [acquisition archetypes](#) and how they can be used to help government programs avoid problems in software development and systems acquisition. Bill, thank you so much for joining us today.

**Bill Novak:** It's my pleasure.

**Suzanne:** Let's start off by having you talk about where the data comes from that you use for doing this kind of work. Because I know we have a very rich set of data sources that inform this work.

**Bill:** That's very true. It's probably best to just start out by pointing out that the SEI does a great number of assessments and evaluations of programs. We often call these [independent technical assessments](#), but they often range even beyond just the technical aspects of the programs. Over the years we've done a number of these. I've participated in quite a few myself and led a good number. The fact that we do these activities on a regular basis means that we get a terrific overview or cross-section, if you will, in looking at acquisition programs of all different shapes, sizes, and varieties. As a result, we tend to see certain patterns that recur across these programs, and we get an opportunity, I think, to look at that and see these patterns in a way that we wouldn't if we were focusing on a single program over an extended period.



**Suzanne:** So, for example, I've just done one of these independent technical assessments. It's anecdotal data for the most part. You get some measurement data, but it's a lot of stories and understanding of the particulars of a situation. So, I have knowledge about one situation. What you're talking about is when we aggregate and look across these, we get a much richer view than just what the anecdotes are from a single organization. Is that right?

**Bill:** It's absolutely true. What we find is that in many cases we'll see some of the same problems plaguing one program that we'd just seen in a previous review that we had done the month before. So, very much of that—we see new problems as well—but we see an increasingly large base of recurring problems across all of them.

**Suzanne:** So, this acquisition archetype is a way of trying to leverage all of that commonality into some presentation that helps people to understand what's underneath it better; right? Tell us a little bit more about what the archetypes are and how they work.

**Bill:** Okay. Well, you know, first of all, let me just make the comment that—many of us being software engineers, software developers, et cetera—we focus a lot of our attention on the building of the software system and the technical intricacies and these kinds of things. That's good, and that's important to do. One of the things we tend to forget in that process is that acquisition programs, themselves, are actually a complex and very dynamic type of system.

So, there are the interactions between all the different people and the stakeholders and the contractor and the program office and sponsors and users and all these different types of folks. We've actually found that a number, a large number of the problems that we tend to see, are actually more a function of the interactions between all of these different groups and stakeholders than necessarily the technical issues themselves, which—although very challenging, in many of these kinds of unprecedented and highly complex systems—are more manageable than simply trying to get everybody to be working together and on the same page.

**Suzanne:** Software engineers aren't typically trained in socio-technical systems and dealing with lots of conflict and trying to get consensus and all of those kinds of things that are necessary for one of these programs, acquisition programs, to function correctly.

**Bill:** That's right. A lot of the issues that would never even become apparent in a very small-scale development suddenly become quite large, and may even begin to seem insurmountable, as you scale up a program to a very large size. So, that's one thing—the programs themselves are these complex dynamic systems. What we've also found is that the systems affect the people operating within them in relatively predictable sorts of ways so that the systems can actually trap us into behaviors, which we may even realize while we're doing them are unproductive or maybe not the greatest idea. Yet, we don't really have a choice, and that seems kind of strange. Let me



---

just give an example in real life. When people see a housing market or real estate market start to go down, people start to think about selling. Well, you don't want to be selling in a market...

**Suzanne:** Down side.

**Bill:** ...that's falling. The hard part, of course, is figuring out when you're at the top and when you're not. What happens is that, when we collectively all see the market start to fall, and we all begin to sell at the same time, the market starts to fall even faster. So, while individually we're making decisions that seem like a good idea—we'll get out before it falls any further, potentially—by doing that we're actually causing the market to potentially even crash. So, that's what I mean about getting trapped into certain behaviors. We see the same kind of thing happening in acquisition programs for the same reason: because it's a complex, dynamic system, involving a lot of people trying to act in concert, but also acting somewhat independently.

**Suzanne:** How does an archetype help us to visualize this kind of a pattern and either avoid it or contradict it, so that we get a more productive result?

**Bill:** Well, the original idea for the archetypes came from work that had been done that was published by [Peter Senge](#) in "[The Fifth Discipline](#)." That was called systems archetypes. So, we're really just adapting and extending that kind of idea forward into our software acquisition context that we're primarily interested in. What we're trying to do is to take the experience we get from seeing this problem recurring in multiple instances on actual programs, and then putting that all together and distilling it down into a very easily read, easily digestible form.

There's a very simple format we have for all these archetypes: we lay out an actual acquisition story of what really happened on a program. Then, we analyze that in a broader, more general sense to explain what was really going on here. Why did this happen? Then, we talk about what are some of the things you could do if you find yourself in the middle of this kind of dynamic, if you will, or this behavior. Or, what could you have done to prevent it? Or, what could you do the next time around to keep it from happening?

**Suzanne:** I've seen some of these archetypes, and they're diagrams. But, they're much simpler than some of the other visualizations that you see in more advanced kind of techniques like system dynamics. They're very readable. They're a way for you to see the people that are involved, the actions that they're taking, and the effects that they're having on either themselves or other stakeholders in the system.

**Bill:** That's exactly right. In some of our research, we've also done work with system dynamics. The whole point with the acquisition archetypes was to keep things as simple as possible and to focus on one problem so that you could kind of tease it out and understand what was really going on in that particular context. Just to back up a little bit, there's a simple way of framing what an acquisition archetype is, which I think is one of the big questions in people's mind[s]. An



---

acquisition archetype describes a situation where an action that's being taken may appear to be sensible and...

**Suzanne:** Rational.

**Bill:** ...promising. Perfectly rational. At the same time, it has unintended, counterproductive effects to what was desired by that action. It might even make things worse than they were in the first place, even though it seemed to make perfect sense. This is the recurring idea or theme behind all of these. We're looking for things where doing what seems to be the most reasonable and sensible thing actually can hurt what you're trying to achieve.

**Suzanne:** So, you're trying to make it visible to program office personnel what some of the counterintuitive effects might be of their actions. Is that another way of saying it?

**Bill:** Absolutely. Because, obviously, those are going to be the ones that are the hardest to even realize that a decision that you make, or an action that you take, might actually end up being counterproductive. So, we can kind of take the cumulative, hard-won experience of many different programs that have run into these problems and distill it down into really a one-page format, which is what we created for the acquisition archetypes. And, then, make it available to folks so that they can become aware and, hopefully, avoid repeating that same kind of mistake.

**Suzanne:** Now, you've already identified [12 archetypes](#) that you've published. So, we know that there are a bunch of patterns that are out there that we've been able to identify. I think that's a tremendous accomplishment to be able to offer those to people in such an easy-to-understand format. But, you've done something else that I think is very interesting and goes beyond just kind of simplifying reading about an archetype. Why don't you tell us about [the videos](#) that you've been producing related to these.

**Bill:** Sure. One of the things that we had found, we had good success with the initial archetypes. When we took them to trade shows and things, they went very quickly. People found them really interesting, but we wanted to be able to try and reach even more people. So, we thought, "Can we take these and make them even more accessible and even more understandable? We've taken two of them so far and turned them into, essentially, videos. Or, they're animated videos that explain pretty much the same information that you'll find in the one-page sheet. They're only five minutes long. They're pretty easy to watch, and they can lay it out piece-by-piece. They can build up the simple diagram we have, and explain as we're doing it, "Why does it look like this? What goes on? And, why is making this particular kind of decision going to, down the line, lead to this unexpected and not very helpful effect?"

**Suzanne:** I will say, I've watched them, and they're engaging. They draw you into the story, and yet they're very simple to watch. They're not something where you feel like you've got to watch



---

it 16 times to get the idea. They are very clear in terms of how they do that. So, I congratulate you on that because that's not easy to do.

**Bill:** Well, I appreciate hearing that, but let me make sure that I give credit to Kurt Hess, our graphic designer and very talented animator, who also works at the SEI. He really did a terrific job coming up with very compelling visual ways of expressing what, in software, are sometimes pretty abstract ideas. It's always a little bit hard to get your hands around software because you can't hold it in the same way you can other, more concrete things.

**Suzanne:** So, what are the two archetypes that you've converted into videos? What topics do they deal with?

**Bill:** The two we've done so far are called, the first one is called [Firefighting](#). The second one is called [The Bow Wave Effect](#). They are both also available in the one-page write-ups, and you can download those off the web. We were just describing two of these dynamics that we thought were especially interesting and that were widely prevalent in our experience in working with actual programs. The first one, Firefighting, just very quickly, talks about the way in which, when you begin to encounter problems—perhaps more problems than you expected in the software—sometimes developers that would have been working, perhaps, on the next generation of the product, get diverted to fixing bugs in the current release. And thereby, you're kind of undermining your ability to develop things for the next release. That's actually based on work that was done by [Nelson Repenning](#) and his team at MIT, and we had adapted it to an acquisition context here. The second one, The Bow Wave Effect, is really talking about the tendency we may have when we're doing kind of a spiral development, if you will, of maybe postponing some of the most difficult tasks, some of the most complicated tasks, that we should do early on, to do the best job in mitigating risk. Instead, sometimes we push them out a little bit, and sometimes the reasoning may be because we can make better progress on these other things if we're already behind schedule as is often the case, even early on in many large-system development efforts. So, these larger tasks can get pushed out in time and then, all the while, risk is increasing. The system is becoming more set in stone, if you will. And so it becomes harder to make...

**Suzanne:** More and more brittle.

**Bill:** ...the changes. Exactly, more brittle. And so, as a result, you are undermining your ability to do these more complex tasks as the time goes by. And we don't do this consciously. We don't do it deliberately. We're not trying to make our lives more difficult. But, there are other pressures, and we can find ourselves in this situation almost inadvertently, and then realize that it's having a major adverse effect on the program.





---

**Suzanne:** So what's next? Are you going to be doing some more videos? Are you going to be doing some more archetypes? How do you decide which acquisition archetypes are actually worth pursuing? I know you've got tons of data, so you've got to make choices. How do you make some of those choices?

**Bill:** We do have a lot of data, and it's difficult. What we try and look at is what are the ones that are most counterintuitive? What are the ones that are most pervasive, clearly, is a major factor also, so that we can help the largest number of programs by publishing that specific thing. We try to guide ourselves that way. Also [we are] looking at what are the areas that are becoming more prevalent in acquisition. For example, interoperability becomes more important. Then we start thinking, can we be addressing some of those issues? Those are the types of things we try to take into consideration.

**Suzanne:** Okay. That sounds reasonable. Do you have enough of these to work on? It sounds like you have your hands full with these.

**Bill:** We do have quite a few on our backlog list. We're just hoping we'll get more opportunities to continue this research and get some more of these out.

**Suzanne:** One of the things that I haven't seen talked about in the literature about these is their applicability outside of, say, the military context. When I look at them, I see analogs to other commercial contexts where you have outsourcing going on and other kinds of things. Would they be applicable in those settings as well?

**Bill:** I think really they are, for two reasons. Number one, many of them are really focusing more on problems that are purely related to large-scale software development. I think anybody in that kind of industry can certainly relate to that. Other areas where we talk about some of the interactions between program management offices and contractors and things may not immediately seem relevant. But, increasingly, as we begin to see more and more outsourcing being done where you're managing an independent...

**Suzanne:** You're not doing the development yourself.

**Bill:** You're not doing it yourself.

**Suzanne:** So, you're doing the same thing that the government does then.

**Bill:** So, essentially, you're doing some very similar kinds of things. And, therefore, you're prone to some of the exact same kinds of situations. So, it really does go more broadly even though we've been developing them with...

**Suzanne:** The data set is primarily military and government.



---

**Bill:** Exactly. But the problems are, in fact, all too universal.

**Suzanne:** I know some people out in the outsourcing world that would probably agree with you on that. Well, we certainly look forward to new archetypes, new videos, and things as they evolve with your work. I want to thank you so much for joining us today. It's always a pleasure to talk with you, Bill.

**Bill:** Thank you, Suzanne. I really appreciate the opportunity.

**Suzanne:** If you'd like more information about Bill's work on acquisition archetypes, or if you'd like to check out the two cool videos his team has developed, please go to <http://www.sei.cmu.edu/acquisition/research/archetypes.cfm>.

This podcast is available on the SEI website at [sei.cmu.edu/podcasts](http://sei.cmu.edu/podcasts), and at [Carnegie Mellon University's iTunes U site](#). As always, if you have any questions, please don't hesitate to email us at [info@sei.cmu.edu](mailto:info@sei.cmu.edu).