Carnegie Mellon University
Software Engineering Institute

SEI Podcasts

Conversations in Software Engineering

An Exploration of Enterprise Technical Debt

featuring Stephany Bellomo as interviewed by Suzanne Miller

Welcome to the SEI Podcast Series, a production of the Carnegie Mellon University Software Engineering Institute. The SEI is a federally funded research and development center sponsored by the U.S. Department of Defense. A transcript of today's podcast is posted on the SEI website at sei.cmu.edu/podcasts.

Suzanne Miller: Welcome to the SEI Podcast Series. My name is Suzanne Miller. I am a principal researcher in the SEI Software Solutions Division. Today, I am joined by my friend and colleague, Stephany Bellomo, a principal engineer in the SEI Software Solutions Division also. We are here to talk about her team's recent work with enterprise technical debt and the risk it represents. Welcome, Stephany.

Stephany Bellomo: Thank you.

Suzanne: This is your first time recording a podcast with us. I know you have done lots of webinars, but let's get our audience a little more familiar with who you are, what brought you to the SEI, and what is the kind of work that you do here.

Stephany: Sure. Well, let's see. I started out a long time ago. I have a master's in software engineering. I started out as a developer developing satellites, actually, for Lockheed Martin. That was a long time ago, and I could not let the internet pass me by. I know that dates me a little bit. But anyway, I actually went to Intuit and did development for them and was on their architecture team there. I eventually joined the SEI in 2007. I was interested in doing more in architecture as well as research. Today I am in the Architecture Design Analysis and Automation group where, over the past few years, I have been focused on some research in technical debt as well as release engineering. I also teach courses at the SEI in software architecture and technical debt.

Suzanne: Yes, we have had lots of fun working together over the years in different aspects of that. Yes, Stephany and I both have had time in the satellite business. That is one of our connections. I am really excited to talk to you today because it has been a while since we spoke, and you have been doing some different things. This newest thing is enterprise technical debt. Start with explaining what is enterprise technical debt, and how is it different from, I will call it *software technical debt,* which is what we have talked about at the SEI over the past several years. Is it just a matter of scale? Or is there something else that differentiates the two?

Stephany: Let me talk a little bit about what is common. With enterprise technical debt, you still are dealing with choices that were made to be expedient or maybe shortcuts that were taken that in the long term actually end up being costly or a lot of effort and things to deal with. You basically have a case where people are making shortcuts, and now you are dealing with this accumulation of debt. That is what is in common.

The difference is that when we talk about technical debt, a lot of times we are just talking about a single-system context. When we talk about enterprise technical debt, we are talking about technical debt where the impact can be felt across multiple systems or multiple parts of the organization. That is a real big difference. It is not really just about how many systems are involved or things like that. It also has to do with the people aspects and the impact across the organization.

Suzanne: Your blog post talks about how you came around to this when you were doing some system architecture evaluations. What I found interesting are some of the things that you found. Tell people, so what were you looking for, and then what did you see that made you realize that you were looking at not the typical issues that you are accustomed to, and that maybe there is this other form of technical debt we should be paying attention to? Tell us how you got there.

Stephany: Sure. When we do architecture evaluations, basically, we are looking for design or development risks. That is kind of what we are doing. We are going about our normal daily work for architecture evaluation and probing for risks and analyzing documents and those kinds of things. We noticed that some risks actually resulted in the accumulation of additional effort, cost, pain. You could see in the consequence of not dealing with the risk or the issue, it was getting worse, and there was evidence that it was getting worse. We know what we have learned about technical debt is that one of the differentiations is that the consequences do get worse, like accumulates over time. We started to see some of those, and so we started to say, *Hey, wait, these are not just risks. These are technical debt items.* So we started to capture those.

As we did that, we started to realize that some technical debt items are not like the others, that some of them actually have a pretty big impact if they are not fixed. For example, one of the ones we saw early on was a case where there were two different systems and two different teams who were both writing to the same database schema directly, which on multiple levels is bad engineering practice, but I will set that aside for a moment. In addition to that, it was also a case where neither team could actually make the change to fix it because it would require actually creating an API [application programming interface] and changes for both sides. Neither one could necessarily fix it, but it was just going to get worse and worse over time.

Suzanne: So there needed to be some kind of adjudication governance activity that was over both of those teams to be able to actually enable a fix.

Stephany: Yes, exactly. That is actually, you just kind of stole my thunder. I was going to talk about that maybe a little later. But I think the big key is the people aspect. I love talking to you about that, Suz, because I know you do a lot of work in change management and the people aspect. But the bottom line is these items also could not be fixed by a single project team. You could not put it in your backlog and just have a team work it off. You have to bring in other parts of the organization.

Suzanne: And there has got to be some stakeholder management that gets involved. When I looked at the blog post, that was one of my insights, is that if I were to paraphrase it, *enterprise technical debt is what you see when stakeholders are not being managed consistently*. You have this promulgation of software-system technical debt but, like you said, it is things that I need to get multiple stakeholders involved. I need to have some different governance structures besides just the team, so that I have to go outside. I have to go to the enterprise level, whatever that is for me, so that I can actually get these things corrected. That may be actually one of the things I was thinking about is, as a barrier to teams actually even identifying technical debt that they know they cannot fix themselves. This concept actually gives them a way to identify those kinds of technical debt items and highlight for their leadership; it is like, *Hey, this is real, even SEI says so*, *so we need to figure out a way to deal with that.* Is that a correct reading of the blog post?

Stephany: Yes, I think it is a definite characteristic. If you see a case where you have a technical debt item, it is accumulating in some way, either effort costs, complexity, something—things are getting worse, so you know those characteristics are there. The additional characteristic is it is impacting multiple parts of the organization, and it will probably take multiple types of stakeholders, possibly at different levels, often higher levels, to resolve, then you probably are looking at a technical debt item for sure.

Suzanne: That is probably one of the biggest differences in how you treat what we would call traditional software technical debt and enterprise, is that need for multiple stakeholders at multiple levels, where when we look at more traditional technical debt, we assume that it is fixable, addressable by the project team that identified it. Is that a correct way of looking at it?

Stephany: It is, but there is a little bit more to it also in terms of how you have to manage it. One of the things that is really helpful is to have a technical debt repository as a best practice anyway. But it becomes even more important for enterprise technical debt because if you have a repository and you are keeping track—especially if you have a cross-portfolio repository that is keeping track of all these technical debt items—you then can see evidence of the same item popping up across multiple projects. And using that evidence, you can then motivate the leadership to do something about it. That is really important too. So it makes having a consistent cross-organizational technical debt repository also a really powerful and important part of managing technical debt, if that makes sense.

Suzanne: I would say that is your first remediation strategy in this, is to actually have a way of seeing what is the debt that is accumulating across the enterprise. What are some other remediations that you recommend to people or that you have found to be useful in remediating enterprise technical debt and maybe how they would differ from strategies you would use for traditional?

Stephany: I think that is really the main one. It all starts with having that repository and building that evidence. So, I cannot overemphasize the value and importance of doing that; and that is a non-trivial thing to set up and to get everybody to do—to use it, to put the processes in place, all of those things. The other practices that go along with that are regular grooming of that repository, having somebody go in and look at it and see, *Hey, do I see technical debt items that are popping up across there that map to multiple projects?*

For example, one we saw was a need for or lack of a common access-control and authentication capability. Everybody was off building their own heterogeneous, and not only did they have security risks related to that, but teams were drowning in the cost of doing it themselves. It was so many. We could see by having that evidence and grooming the backlog regularly, we could see, *Hey, it is showing up on this project*. It came from all these different architecture evaluations. We could see that same technical debt item pop up, pop up, pop up. We took that evidence, and we were able to convince the leadership to invest in the shared access-control capability. The combination of having a key at the process side of it and also the repository side of it, and then you also need to have a forum to bring it to your stakeholders. You need to create that connection to the next level up, which is also like you were mentioning, that is a governance thing. You need to put that in place. I think one other thing I could say is that it is important to have good facilitators like yourself.

Suzanne: Oh, thank you.

Stephany: This is one shameless plug: Reach out to us if you want to have facilitation help on any of this, because a lot of times, people are just stuck. You are stuck at the project level suffering the pain, or you are stuck at the leadership level paying for the cost of all these people who are doing access control themselves over and over again. There are organizations living with the pain, but nobody can really talk across them as an independent.

Suzanne: Yes, and one of my mantras for a lot of different kinds of repositories is you do not want a write-only repository. That becomes another millstone around the neck, and it becomes something that is perceived as wasted effort. So that use of the repository to analyze, refine, understand, and take action. The staff seeing that you are visibly taking action on things that they wrote about is a very powerful reinforcer for them to be actually using that repository to actually add what they know.

The thing that is important about that is, it is at the team level that you know the thing, whatever the thing is, like the common access-control issue. If none of those teams wrote about it, then nothing would get done about it. Management would not even necessarily until they saw licensing costs for, *How come we have 16 different tools for doing authentication?* Until they saw something like that, they would not even know there was a problem. Part of our role as developers is to give that information to our leadership, so that they can act on our behalf. A lot of times, we lose sight of that. I am reinforcing your point on that. I just named one of the obstacles of doing this, is that people see it like it is a waste of time. *If I identify these things, nobody is going to pay attention.* What are some of the other obstacles that you have seen organizations need to overcome when they are trying to implement approaches to this?

Stephany: I think the people side of it is really a tough part. Building that bridge to be able to have projects communicate and capture these issues and communicate them upward. A lot of times…In fact, I was just teaching this technical debt class a few months ago, and we got on the topic of enterprise technical debt at a break. I was telling them about it, and they were like, right away, they said, *Oh, my gosh, we have so much of that.* They just started spewing all these stories about it, all the examples. I thought, *Well, what are you guys doing about that?* and they are like, *Oh, well, we cannot, I mean, we have no…I am like a developer that is below this manager, below this manager, and the person that we need to fix that is, like, way, way, way up at the top and doing model-based engineering problems.* They said, *There is no communication path for us to actually escalate that up without looking like we are being insubordinate or something like that, and there's no funding.*

The other big problem is there is no funding. So you are at the project level. You are getting your funding there. Who is really going to pay for that? A lot of times at the enterprise level, there is authority, but not a lot of money. Those are troubles figuring out, what is going to be the acquisition strategy or the cost strategy? How are you going to save money for this? Are you going to tax different projects or do a collective, I should say positive, a commune and collect money for it? How are you going to deal with the money side? How do you deal with empowering developers to speak up? Those are tough problems.

Suzanne: We have data about the benefits and the cost reduction in the end, or the cost savings, cost avoidance that can be achieved with software technical debt. Have we been able to publish anything of that nature yet? Or are we still at the anecdotal collecting stage? Because I know it is challenging to collect those kinds of measures. Do we have anything to offer people that are saying, *Well, how do I help my management to swallow the cost of this?*

Stephany: Are you talking about calculating the costs?

Suzanne: In demonstrating that addressing enterprise technical debt has a return on investment.

Stephany: Oh, OK. In our experience we have been more successful…. At the enterprise level, the real concern, a lot of times, it is cost, and it is also risk. We have been pretty successful in being able to demonstrate that there is a pretty big risk if you do not deal with something, and that is actually often a more important motivator at the enterprise level than cost. It is important to save money, but if you have an access-control situation where things are completely heterogeneous, you also have a security situation. We find that it is easier to motivate action, and maybe more correct in a lot of ways, to deal with it at the risk level instead of arguing over the money side.

Suzanne: OK. Fair enough. What kinds of resources does the SEI have available or available in the technical debt community for people that are saying, *You know what? I think I have this problem and I need to transition, either I need to get involved in technical debt period, or I need to transition from project-level to enterprise-level technical debt.*

Stephany: Yes, we definitely have resources available here that are experienced with doing those kinds of things. We can help with facilitation as people like yourself and others are great at facilitation here. We have lots of technical expertise to come in and help you identify risks as well as technical debt on your project. In enterprise technical debt, we have experience here with being able to deal with that, so we have staff available. I will be honest with you. We are small. I mean, we are not a huge, huge organization, the Software Engineering Institute, but we definitely have I feel like a very strong niche in this. There is a book, *Managing Technical Debt: Reducing Friction in Software Development*.

Suzanne: I was wondering how long it would take you to mention that.

Stephany: I was not sure if that is what you meant by *resource*. I am sorry. Yes, there is a great book out there. It has nice guidance and helps give you a good understanding of how to differentiate technical debt from other kinds of issues in your code, etc. The book is a great resource as well. There is more information on the SEI website, and then we have the authors of the book here at the SEI, as well. Ipek Ozkaya and others, so we have nice resources here to help.

Suzanne: The message is you do not have to be on your own, and you do not just have to watch YouTube videos. We have a mix of resources that are self-serve and are also in-person. Do not hesitate to reach out to the SEI if you need to do that because I know this group, and they love challenging situations. I am setting you up, aren't I? But there are lots of different contexts for this emerging and being visible. As we are learning about…I think you guys are still learning the full context of how this manifests and how to deal with it. So, [I] definitely encourage people getting in touch. And, of course…

Stephany: Sorry to interrupt. We would love to also have more examples. That is another thing I forgot to mention, because during the last class I taught, I got more examples, and getting examples, everyone I talked to. The more examples we have, the more we can better characterize it and also make the case that it is a thing, and it needs to be dealt with in the leadership levels of the organization so that the developers can stop suffering down there at the project level.

Suzanne: No, no, no. Exactly that, yes. This has come about because of SEI research activities as well as customer-engagement activities. I know you, Ipek, all of that crew. What is coming next in the research aspect of this that you guys are looking at?

Stephany: Ipek is doing some exciting work on the technical debt side, so we will refer you to the website for that. For the enterprise technical debt side, we really want to collect more examples and provide more guidance and that type of thing on how to manage technical debt.

Suzanne: OK, and I look forward to having some other conversations with you as that emerges because these are always fun for me to update on things that I do not get to talk about on a regular basis. So this has been really great. Is there anything else that you wanted to tell our viewers that we have not talked about enterprise technical debt, either cautions or, *Please be sure and think about this*, or we have enough information at this point to send out to them?

Stephany: I think we have covered most of it. If I were to say anything, starting by creating that repository, starting to look for it, and using the book is a great resource for you to understand what is a technical debt item and how do I capture that in my repository? What fields am I capturing and putting? That practice really is so simple but foundational because what you write down, what you make visible, you can fix. That would be my recommendation for starting anything in technical debt.

Suzanne: Yes, you and I are aligned in so many ways. Make implicit things explicit. That is the first step. I want to thank you so much for talking with us today. We will include links in the transcripts to the book and other resources that we talked about, as well as the two recent blog posts on enterprise technical debt that are out there. I want to remind our audience that our podcasts are available everywhere. SoundCloud, Stitcher, Apple, Google, and of course, the SEI's favorite, the YouTube channel that we have on YouTube. If you like what you see in here today, please take a look at some of our other offerings. I want to thank Stephany and I want to thank you, our viewers, for joining us today.

Thanks for joining us. This episode is available where you download podcasts, including SoundCloud, Stitcher, TuneIn Radio, Google Podcasts, and Apple Podcasts. It is also available on the SEI website at sei.cmu.edu/podcasts and the SEI's YouTube channel. This copyrighted work is made available through the Software Engineering Institute, a federally funded research and development center sponsored by the U.S. Department of Defense. For more information about the SEI and this work, please visit www.sei.cmu.edu. As always, if you have any questions, please do not hesitate to email us at info@sei.cmu.edu. Thank you.