Mission-Based Prioritization: A New Method for Prioritizing Agile Backlogs

Featuring Keith Korzec 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: Hello, my name is Suzanne Miller. I am a principal researcher here in the Continuous Deployment of Capability Directorate, with my colleague and my boss, Hasan Yasar. He has been a frequent guest of ours on the podcast series, and I want to welcome him back. So, welcome Hasan.

Suzanne Miller: Hello, my name is Suzanne Miller. I am a principal researcher here at the SEI, and I would like to welcome my colleague Keith Korzec, who I have worked with for many years. He is also a colleague of mine in the Agile Acquisition Transformation group, and he works with many DoD [Department of Defense] customers across the Navy, Air Force, lots of DoD elements. He is here today to talk to us about a new method that he is using with customers called Mission Prioritization [tool download], and it is a different way of prioritizing an Agile backlog. As we know, as we move into Agile contracting, backlog management is one of the big rocks that we all have to deal with. Keith is here to talk to us about some of the things he has done to help with that. I want to say welcome, Keith. Before we get started into that topic, tell us a little bit about your background and what is it that drew you to the SEI and this kind of work.

Keith Korzec: Well, thanks, Suz. Pleasure to join you. My background a little bit is the…I had spent 20 years working for the government, started out with the Navy, and worked a lot of software for fighter aircraft, later became part of Defense Logistics Agency, and then later on the DCMA, Defense Contract Management Agency. I have done a lot of work in the acquisition side. Currently, I am supporting the training and coaching of program offices that are adapting…their contractors are adopting an Agile development methodology.

Suzanne: And some of these are pretty big, so they are not just little tiny ones, right?

Keith: Yes. We will talk a little bit about that more in a second.

Suzanne: Yes, because that is really where the problems come in. So, this is a method, mission-based prioritization, and it is really geared towards large-scale Agile development efforts. So let us talk about why that matters, and why that is a problem for these large programs.

Keith: In my experience, most of the programs have attempted to do a weighted shortest-job-first method to prioritize. What we have found in reality as the issues come up is that that method is financially focused, and it is very time intensive. A small program with only 100 items in its backlog would take 400 comparisons just to prioritize that. When we start to consider that size of a program, it is barely manageable, but a medium-sized program I am currently supporting has 178 portfolio epics, 467 program epics, over 200 capabilities that have already been decomposed, and over 900 features that are in their backlog as of current state. And it is evolving larger and larger. So, the comparisons…

Suzanne: So, this is really unmanageable. The method that you talked about, Weighted Shortest Job First, is a method that is recommended in Scaled Agile Framework and other frameworks that really leverage Lean, and it is a bilateral comparison, or it is a comparison against a norm, right? We call it *Where is the chicken*? and then you compare everything to that. As you said, the number of comparisons makes that method really tough to do on a large scale. That is really where your method is very interesting to me, because you start to deal with that. So, tell us how you deal with that issue.

Keith: Right. One other aspect of that is, every time you add an item or take something out, in theory you are supposed to do that whole comparison again, the entire backlog. So that becomes totally unmanageable.

Suzanne: Right. So, tell us what is different about mission-based prioritization. How does it solve that problem of the unmanageable number of comparisons?

Keith: What we have done is, we have set up a list of 10 parameters that are mission focused. The team that is doing the backlog prioritization goes through one time, and weights them against each other. So those weighting factors are applied. Then the entire backlog is looked at one item at a time as opposed to comparison against each other. Does that factor apply, or does it not? And with the tool that we have created, it just does all the prioritization in the background and gives you a list of, based on the prioritization. And that is a starting list, it is not the final list. Things can be moved around, but it does cut out a lot of the animosities and the arguments that happen during normal backlog grooming of, *What do I think or you think?* on the most important.

Suzanne: Give us an example of a couple of the different parameters that are in this list. You do not have to give us all 10, but just a couple of the different parameters so people can have an idea of what are we using to help with this prioritization.

Keith: Some of the items we consider for the mission-based prioritization are, are they fundamental to other work that is being done? Does it provide a tactical advantage over an enemy? Does it deliver a new capability that does not exist? Is it fixing a deficient capability that is already fielded? One thing to mention, though, is that the tool is completely customizable. So, if those do not apply, or you have others that would apply better or not apply at all, you can modify that any way you want.

Suzanne: So, this is taking the complexity and containing the complexity in the comparison of the factors against each other, which is a thing that you do with stakeholders in the room that really have the mission focus. And then you get that done one time, or you may do it once every six months or once every release, but you are not doing it every time a new backlog item comes in. So that is really one of the efficiencies you are using.

Keith: Right, and as long as the parameter does not change, there is no reason to go back and evaluate it, because you did a binary assessment—does it apply or does it not? So, adding new items to the backlog, or pulling some out, or moving them to later does not affect the outcome of them.

Suzanne: One of things that I find with tools like this is that they actually help us with non-intuitive answers. I think that this is going to be higher priority than that, but then when I apply all these things, that really increases. Are you seeing that kind of effect from using this scheme, and when you see that, does it help people to understand the system better? Or, is it just a surprise, and I do not know what I am going to do with that?

Keith: What I have encountered in a couple of these pilot programs is after that first run of a dozen or so items, we step back and take an assessment on it, and say, *Does this make sense?* If it is out of alignment with what the group believes, they go back and look at those factors, and then they realize that, *Hey, we weighted this one way more than we thought we should have*.

Suzanne: Got you. So, you do some validation of the model itself after you have done a small run, to make sure that it is really consistent with what the stakeholders that understand the system really think are the important areas.

Keith: Correct.

Suzanne: OK. Can you talk about any of the effects that this has had in terms the pace of backlog refinement or other things that you have seen come out of this that have increased the effectiveness of Agile adoption in the settings you are working in?

Keith: Sure. Most of the programs I have dealt with—I would almost say all of the programs—we went through the backlog-refinement process once in the old method, and they avoided doing it again. They did not want to address it, it just took too much time. In one program, it was four days, four full days of doing that to get through about a quarter of the backlog. With this methodology, once we started implementing this into several of the programs, they added this to their weekly meeting. So, any new feature that comes along, they say, *Hey, we have three features. Let us run through this*. Bam, bam, bam, and it is done. And the current backlog is re-prioritized.

Suzanne: Excellent. So that was to the point of iteration and incremental that people are willing to do the iteration, because the method is easier, and it gives you a good enough result that people are comfortable going forward with it.

Keith: Correct.

Suzanne: OK. If people are interested in this, and I can imagine some of the customers that I know are interested in doing something simpler for their backlog management, how do they get hold of this? What do they need to be able to start using this method on their own?

Keith: We have written a blog [post] that explains the background, the justification, and gives detailed instructions on how to use the tool. And within that [post], there is a link to the actual tool that can be downloaded. We do have two versions of the tool. One has no macros, no scripting language. That is good to be used on government computers. There is another one that has some minor script that makes it a little easier to print out the prioritized list. And that is also available off that link.

Suzanne: Depending on your computing assets, and we always have to worry about that. So, this has been something that I know you have been passionate about, in taking the Agile dogma and figuring out more pragmatic, practical ways of applying those principles in these very complicated settings. This is an example of that passion. What are some other things you are thinking about? What are you looking at now that may be something that six months from now we will get a chance to talk with you again about what your latest creation is? What are you thinking about?

Keith: The latest hot-button issue we have going on within our team is implementing earned value management on Agile programs, and we are working on an FAQ and a list of items that

need to be decided on prior to starting the program, so that you do not run into trying to force fit it into an existing EV [earned-value] system.

Suzanne: Again, I know customers that are going to be on the edge of their seats waiting for that. I am glad that you are taking this on, because it is something that…Agile has matured enough out in the space where we actually have some data, and we have some things to say about, *If you do it this way, it is not going to work as well. And you probably want to look at these other things.* So, I am glad you are working on that.

Keith, thank you so much for joining us today. As always, the resources that we mention, like the blog posts that you talked about, are available on our website. Go to insights.sei.cmu.edu. Easiest way to search is on your last name, Korzec, which is K-O-R-Z-E-C, and that will also lead you to the link that you mentioned in terms of the actual downloadable tool. So, those are things that our viewers can access. As always, we invite you to ask questions via our info link, info@sei.cmu.edu. For all of our viewers, thank you for looking at our podcasts, wherever you get them. We look forward to talking to you in the future.

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