From Process to Performance-Based Improvement
featuring Tim Chick and Gene Miluk interviewed by Suzanne Miller

----

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 at Carnegie Mellon University in Pittsburgh, Pennsylvania. A transcript of today's podcast is posted on the SEI website at sei.cmu.edu/podcasts.

My name is Suzanne Miller. I'm a researcher and senior member of the technical staff here at the SEI. Today, I am very happy to introduce to you, very good friends of mine and colleagues, Timothy Chick and Gene Miluk, both of whom are fellow researchers here at the SEI. At the SEI, Tim and Gene work with both commercial industry and government organizations in helping them improve their overall performance based on the organization's goals and objectives. They are responsible for defining, developing, and transitioning into practice high-performance software and systems engineering practices based on Agile principles. In today's podcast episode, we will be discussing how to move from process-based to performance-based improvement. You both know this is a topic that I fully support and am engaged with in my own research as well. So, I want to welcome both Gene and Tim.

Tim Chick: Thanks for having us.

Gene Miluk: Thanks, Sue.

Suzanne: All right, so, for those of our listeners who are just embarking on some kind of improvement effort, would you explain the difference between process-based improvement and performance-based improvement? They both begin with P, so that could be confusing.

Gene: Tim, I'd like to take a stab at that. I've been at the SEI maybe too long, 21 years and so has Suzanne. So, I'm not the only one.

Suzanne: We started about the same time.

Gene: Started about the same time. Tim is a brilliant, young engineer and he often challenges me with, *Why the hell would you do it that way?*

Tim: Sometimes you have a good answer.

Gene: Once in a while, I can remember that far back, actually. Well—when the CMMI Institute took the CMMI, which I spent a lot of years working on and developing, actually—Tim would ask, well, *the DoD wants you to do something new, something innovative. So, what's the next thing*? And, we're struggling with *what's the next thing*. Then, Tim came by and said, *Well, why did you do process improvement to begin with?* So, I started thinking back and looked at the environment in the early '90s and I would go to customers and we would look at best practices. Remember Watts [Humphrey's] best practices?

Suzanne: Oh, yes.

Gene: And, I can remember talking with senior software people and saying, *You need a software development plan*. And, they'd say, *Why? I have a project plan. Why do I need a software development plan?* I said, *Where's your configuration management?* They would say, *We're controlling our source code what else do we need?* I realized that back then an infrastructure wasn't in place because I did measurement at the time. I remember talking with groups and saying, *I get more metrics out of a little league baseball game than I got out of a multi-million-dollar software effort.* So, the measurement systems weren't in place to get to performance. So, what did we do? We looked at those companies that had good processes in place, and we said, *If you were to mimic those processes, you would get good performance.*

Suzanne: You are more likely to get good performance because there is more than your process that's going to determine that.

Gene: At the beginning, I said, *You shouldn't do this totally. You should measure things, but most of them didn't*. They said, *Well, I want to get better. So, if I look at people that do things well, and I imitate them, do what they do, I would get better*. So, we would measure performance, a proxy of performance would be our practices. So, *how do we know we got better? We started level two and now we're level three.* Well, we didn't measure performance.

It was kind of interesting. I was just thinking back about that time—does anybody remember the Fosbury Flop? If you're in track and field, when you did the high jump you would normally jump straight over with your face looking down. There was this young man out of California that did a backflip over the high bar, and he revolutionized the high jump. Why? Because he kept winning matches. So, if you start looking at performance improvement versus process performance—if you were a golfer, and you could have a textbook swing but never hit a ball—that's process improvement, okay?

Suzanne: That's not a terrible analogy.

Gene: Well, this is an informal one. The point is—and Tim kind of inspired this because when we go out now and we talk to organizations— a lot of that infrastructure is in place. You don't have to say, *Why don't you have a project plan for software?* They do. You don't have to say, *Why don't you have configuration management?* They typically do.

Suzanne: Well, and if you go back—I'm going to jump in here out of my interviewer role into some of my history. Those models were based on the fact that when we looked at the software industry in the '80s, and we compared it to the manufacturing industry, which is where some of these principles came from. You look at [Deming principles](#) and some of the things that came out of the [total quality movement](#), they had basic practices in place that allowed for the measurement. As you were saying, the software industry didn't. So, those models were meant as a surrogate for things that were missing from the environment to be able to do performance-based improvement. Theoretically, when you get to the higher levels of those models, you're supposed to switch over to performance-based improvement. Sometimes it happens; sometimes it didn't.

Tim: What we're finding is a lot of organizations now have the ability to have configuration management. They have tools that are actually collecting data.

Gene: Let's go through the defect tracking they have. They have some sort of time accounting or project-management system.

Suzanne: Estimation.

Gene: Development environments very often have that information in there. So, there is much more data than 20 years ago.

Tim: But, they're not using it. That's what we're finding.

Gene: Well, that's true.

Tim: What we're finding is different people actually have different pockets of data, but they're not coming together and actually using it for performance.

Gene: I want to claim success where I possibly can. I think one of the big successes of process improvement was that those things are now taken for granted. You're going to have an environment that collects that data. You're going to have a certain set of tools, testing tools, defect tracking tools. All of that is now expected to be in place that wasn't 20 years ago.

Suzanne: Although I have to tell you there's still some environments I run into that are not quite as advanced as that.

Tim: I went to one customer. Why did they call me? They didn't have any source code. They lost their source code. I actually worked with a company who lost their source code, not to one product but to multiple products. They didn't realize it until one of their customers came back and said, *I would like you to update my product*. And, they couldn't.

Suzanne: There's still some gaps.

Tim: There's still some gaps, but, by far, that's the exception. The majority of mainstream have accepted that certain practices are just fundamental good engineering practices in software.

Suzanne: That's true. I guess what I want to say is that I see them as two different lenses. There is the lens of process, which is a lens that is about looking at behavior of individuals and groups. Then there's the lens of performance, which is looking at *what is the outcome of all that behavior*. So, in my mind, the performance-based-improvement focus is one that is really trying to integrate the behavior with the goals and the outcomes. That, to me, is the new thing. We're not divorcing—we used to say *process versus product*—we're not doing that. We're saying, *No, you have to have enough process so that you can pay attention to the product.* But, process without goals and outcomes attached is going to get you the golf swing without a ball.

Gene: So, let's go with that a little bit. It's something that breaks my heart as a software professional and a researcher. I go to many software executives and I say, *What's important to you? Is quality important to you? Yes.* Then you ask questions: *What's your defect density? I don't know. What's your cost of quality? I don't know*. It's embarrassing, and it doesn't go anywhere. What I find so amazing about this process, the checkpoint diagnostic, is that—just as Tim said—they are collecting this data, but they don't know what [sic] to use it for. So, if you can use the data they have and come up with defect densities and compare it, all of a sudden light bulbs go off. If you can look at rework and cost of quality and show it to them on their own site, light bulbs go off.

Tim: To go back to the process part, you also map it back to say, *What practices are you not doing that are considered best practices?* Then, if you use some historical database on industry, you might even be able to project the return on investment of actually adding certain best practices that are missing with their own data. That's how you actually set performance-based improvement goals. You try to implement something, and then you can re-measure it using the data that you already have. You can collect and re-evaluate.

Suzanne: You mentioned the checkpoint diagnostic tool. I want you to tell our audience what is that and how did that come about? What is it being used for in this overall performance-based improvement effort?

Gene: Let me take a few minutes to describe the flow of the checkpoint diagnostic. It's a very short, inexpensive diagnostic compared to a full-blown appraisal. It starts out with a dialogue of,

What's your business goals and objective relative to the software? What's important? Is it high quality levels? Is it cycle time? Is it delivering features? What is your business value to software?

Then, what you do is you map that business value to a best-practices model. If quality is important, what are the quality practices that help you get there, in some standard model we like to have? OK, a best practices model. That's half of it. The other half of it is—and I have to really give Tim credit because when he proposed this, I kind of thought he was crazy. He still might be crazy, but this time it worked.

Tim: You told me it wouldn't work. There's no way they'd be able to do it.

Suzanne: Never tell Tim that.

Tim: I love a challenge, don't I?

Gene: As you go through the best practices model on one side of the ledger, on the other side of the ledger you're keeping an inventory, creating a data model: *What tools do you have? What data's in the tools? What records do you keep? What data is in the records?* You create a big inventory of those, and you have them show it to you. What you find is that most places don't know all the data they have. Someone knows about the defect tracker. Someone else knows about the project planning. Someone else knows about configuration management.

Suzanne: Tim's got a spreadsheet over here that nobody knows anything about that has data for the last 5 or 10 years.

Tim: So, the next step is--once you do that--is to say, *Okay, each of these individuals or each of these different departments have certain pieces of data.* They all align up with certain aspects of your roadmap. Maybe it's a release. Maybe it's certain milestones. You say, *OK, I want all of this data. And, by the way, this person has this data, this person has this data, and I want you to align all that data with these milestones or these release events, right?*

Suzanne: Phase-gates.

Tim: Phase-gates. However you want to do it. Their business structure will dictate what that is. Then you take that, and we really do a qualitative analysis where we ask, *OK, what does this data mean?* What is their defect density, if they have size and defect data? What is their return-on-investment? How much are they spending? Are they hitting the schedule? Time-to-release, Right? Those types of things. Then we report that back to them because we also did that qualitative piece, that model-based, traditional practice-focused one where you can go back and say, *Here's what your data says. Here's how your data compares to some industry because we do have lots of industry data.* Then we say, *Where are your weaknesses?* Well, these types of weaknesses usually align up with these types of practices that are either non-existent, or kind of

weak, in terms of your implementation. Then, management executives say, *OK, if I focus some energy here, I would expect these numbers to improve.*

Suzanne: Right. There are certain practices that do have pretty strong corollaries to certain improvements and performance.

Gene: In certain environments.

Suzanne: In certain environments, that's right.

Gene: Measurement, you don't know.

Suzanne: Environment is all, yes. But, you can make some of those assertions in the environments once you've got the whole context. That's a much more powerful—in my mind—argument for doing this than, *Model X says we should do this.*

Tim: So, it's more encouraging to people to actually change their behaviors. They have numbers in which they can actually determine whether they've actually improved. That's the biggest thing I found going out with doing the model-based improvement, right? Well, we now do the practice. Great, are you a better organization because of it.

Suzanne: There was a guy at Boeing who's now at CMU who—when he would receive appraisal results, process appraisals from any of his suppliers or his organizations—he demanded certain data and performance. He was the only one I knew that did that. He had that sense already that—if you just do the practices and you don't see improvement in your performance—I'm not getting a good deal on this. There's some precedent for this just not in…

Tim: I used to always tell a new process improvement group that says, *We are getting ready to introduce some new initiative. We're going to do some improvements.* At some point your sponsor is going to say, *OK, Tim we just spent half a million dollars on this process improvement initiative, what did I get for my money?* The time to prepare to answer that question is not when asked it; it's now, before we start doing the initiative. That's the only time we're going to get the baseline data to be able to try something new and then say, *OK, did I actually improve or was it just different*?

Suzanne: Tell me a little bit about what you've seen so far in using it. What kind of context have you been using the diagnostic in? What kinds of results, either in terms of motivation or in terms of actual performance, that you can talk about that you've seen so far?

Gene: I'll talk about one example and let Tim talk about another one.

This is really short and sweet, but it condenses a lot of data-collection analysis. We had one organization that was in a regulated industry, but the software was not in their product. Their

software was a utility product that the customers used to make their product more useful. I can't say anymore about it than that.

We went in, and we did the diagnostic. What they wanted to do was put more features in their product so that their customers would have more value than the competitors, OK? Because the product they were selling is actually a commodity product. You can get it from multiple places. So, the software utility created…

Suzanne: That was the value ad.

Gene: …better advantage, right. What they found is, their marketing people said, *Hey, we need to put more into this so we can get more value to our customers.* And we were able to. So, we went through the checkpoint diagnostic, concentrating on those sets of issues. Long story short, we filtered through and had a couple of deep dives in the data. Some of it was actually in manual form of format. When we looked at it, the last release was 7,000 hours. They found 1800 hours of rework in the requirements phase of development. When we looked at the processes, we saw some issues in RD [requirements development] especially, and we mapped those two together.

Now, that is pretty much a no-brainer that says, *OK, you have 7,000 hours total and 2,000 hours were in rework in requirements. What do you do?* You beef up your requirements and drive it down. You could make a very compelling business case, value proposition for adding resources there to get more…

Suzanne: More value.

Gene: …out the other side. So that was, you know, pretty dramatic no-brainer.

Suzanne: What about you, Tim? What's your example?

Tim: So, the sample I have is a company called us at the SEI and asked us to help them with some process improvement. What we proposed to them is that we start off just doing this check point diagnostic. Let's see what your problems really are. Then we could figure how to fix it, how to address it. This is another regulated medical device company. The first question we asked, W*hy are you calling us?*

Suzanne: Where is your pain?

Tim: Where is your pain? For them, they were missing their delivery dates. Even worse than that, what really was causing pain is that their customers were reporting defects in their product that was requiring them to do patches to their products. So, their customers were constantly having to reinstall their product.

Suzanne: In regulated environments that means new certification…

Tim: New verification. It was very costly, very painful. They were actually in the cycle for about two years, and they weren't delivering any new features. If you are at the top of your field, your competition is always nipping at your heels. If you're not going to continuously innovate, they are going to catch up. So, they had double pain points. What their data really showed was, yes, they had a quality issue. Based on the amount of defects being found by their verification and validation phase, they could actually predict how many defects a customer was going to complain about. They didn't realize this, but once we did the check point…

Suzanne: You could see the correlation of the data.

Tim: We had the data, and it was a pretty strong correlation. It basically said, *For every so many defects found here, you are going to find that many there.* They were just in awe that, *Oh, we actually have leading indicators on our customer's experience or bad experience that verifies it.*

The other part was project planning, project monitoring, and controlling. They really weren't using real measures. It was all engineering judgment, a lot of guessing. Some of that data was kind of week that it did provide.

We analyzed it. We proposed some improvement ideas. They decided to use the Team Software Process, another tool that we have in our toolkit. So, that really took a more quantitative, more measured, focused, self-directed type teaming environment. That combined with the verification, validation and some of their testing features and beefing some other things up. At the end, they basically cut the number of defects being reported in terms of lines of code of…

Suzanne: As a size surrogate.

Tim: …as a size or surrogate, they cut them in half. They also, for the first time in quite a while, they're now delivering new features, new capability to the customer.

The biggest part is, in all of that, they went through an acquisition where they got bought by another company. Usually, what happens there is you have to justify everything all over again. That data really became valuable to answering that question. *I see here on the books that you spent so much money for these people called the Software Engineering Institute. What did we get for it?*

They were able to answer that question and say, *Here's where our pain was. We weren't being innovative. We weren't delivering products. Here's what our data said about us. We did this improvement initiative. It took about a year. We redid the evaluation based on new releases delivered to the customer. Here's our new numbers. You can see the contrast for yourself.*

That's really some of the power of using performance-based and not just relying on practice-based improvement.

Suzanne: Whenever you have a tool like this diagnostic, what happens is you start seeing new gaps. You start seeing new things that we need to research as the SEI to help people to get better performance. Have you seen any of those? What kinds of things do you see as the future research in this area?

Gene: If you don't mind, I'll go first. I think that things we're looking most intently at integrating into this is the architectural principles we have here at the SEI. We're known around the world for that. We'd like to make sure that once they get the basic pain points in place, there's advantages to design architecture, product lines, and so forth that need to be put in there.

Tim: Some of those things were not really covered by any of the current models. We've been pretty much using the Capability Maturity Model Integration (CMMI). You could use some others that are out there. There's the Software [Engineering] Body of Knowledge, for example. You could go to other sources, if you don't want to use the CMMI. A lot of them don't take the deep dive or the large breadth that the CMMI does. Even the CMMI, it is weakest in terms of architecture and some of the other more engineering-based capabilities.

Suzanne: Excellent.

Gene: We'd love to integrate more of our acquisition technology.

Suzanne: Gene is looking pointedly in my direction. I will just tell you that for those of you who can't see him, since that's one of my big areas of work.

Tim: My other answer to that question would be is I'm hoping we could do more of them. Then, we'll have more industry data across multiple industries that we can begin doing more performance statistical and actually…

Suzanne: More quantitative, empirical analysis.

Tim: Yes, to better benchmarks that are out there.

Suzanne: Excellent. I want to thank both of you for joining me today. This is a wonderful conversation and I hope the rest of you enjoy it too, as much as we did. For more information about the SEI's research in this field, please visit www.sei.cmu.edu/process/. It hasn't changed to performance yet.

And, click on the research tab. This podcast is available on the SEI website at sei.cmu.edu/podcasts and on Carnegie Mellon University's iTunes U site. As always, if you have any questions please don't hesitate to email at info@sei.cmu.edu. Thank you