# Architecting a Financial Trading System with TSP

*featuring Felix Bachmann and Jim McHale interviewed by Shane McGraw*

**Shane McGraw:** Hello and welcome to the SEI's podcast series from the Carnegie Mellon Software Engineering Institute. The SEI is a federally funded research and development center at Carnegie Mellon University in Pittsburgh, Pennsylvania. To learn more about the SEI, please visit our website at www.sei.cmu.edu. A transcript of today's podcast is posted on the SEI website. My name is Shane McGraw and today I am pleased to introduce you to Jim McHale and Felix Bachman who are both currently senior members of the technical staff here at the Software Engineering Institute. Felix works in the Research, Technology, and System Solutions Program, on both architecture-centric engineering and product line practice initiatives. Jim works in the Software Engineering Process Management Program on the TSP initiative. In today's podcast, Felix and Jim will be discussing a project in Mexico that they were both working on for Bursa Mexicana de Valores. BMV is the Mexican stock market and Bursatec is the IT arm that does both the development and the operations of the systems that run the Mexican financial markets. Welcome, Jim and Felix.

**Felix Bachmann/Jim McHale**: Howdy.

**Shane:** Okay, Jim, we're going to start off with you and Felix, feel free to chime in as we go along. If you could give us some of the background on the projects and how you guys became involved with Bursatec.

**Jim:** Okay, so as you mentioned BMV is the Mexican Stock Market; Bursatec runs the financial trading markets, both the stock markets and the futures and options markets, and they're really three separate systems there. Back in early 2009, and probably before that even, they were recognizing that the systems were getting old. They were originally designed and implemented back in the 1980's, so they were really reaching the end of their life and had to be replaced. The other thing that's been happening with the current financial markets worldwide is, there's a lot more interest in Latin America in general as [an] emerging market.

The Mexican Stock Exchange has seen a huge increase over the last few years in their volume. That, combined with the fact that their systems are older, led them to the conclusion they had to replace the system. So they did a build-versus-buy analysis and concluded that they would be better off building their own system from scratch, which was obviously going to have its own set of problems. While they had been doing some new development work in Bursatec, they had not done a program of this magnitude, of this scope, since the original automation of the stock markets back in the 80's. Of course the people who had done that were either gone or a few of them were in senior management but certainly not developers with current skills. So they had that problem: how do you put this system together? Fortunately for us they came to the SEI for parts of that solution, which is the architecture practices combined with the Team Software Process practices.

**Shane:** Now, this was the first time that TSP and Architecture-Centric Engineering (ACE) were used together, so how did that come about? How did you know it was going to work, or was this something that—this was the pilot, and now we know it works, and how did that all come about and how confident were you that they would mix well?

**Felix:** Well, that's actually a good question, how that came to be. It definitely was the first time. So we did not have any evidence, any clear evidence that it actually would work together. It was just driven by the necessity to help that customer. We could not say "you have to have either/or." To be successful they had to use the architecture practice because they had to implement their system from scratch. They had to design the system as well as having a mature, proven process to guide the developers in going through. So it was just a necessity that it had to be that way.

Of course we were confident [though] we never did it, but we were full of confidence that it would work on the problem. If you look at both methods, they have the same goals. One from more the management perspective, so TSP, and the other the ACE practice is more from the engineering perspective. It's the goal to deliver the best possible the system that there is under the given circumstances.

**Jim:** I also think that there was an element of the customer being fairly technically savvy and knowing a little bit about the architecture practices, I guess from coursework, and reading the information they had available on the website, and I think being a little bit familiar with the Mexican initiative that we've had in the TSP over the last few years and saying, "Well, why not? We can't see any reason they shouldn't work together, SEI, do you know of any reason?" Of course, "No, we think they'll work together." So again, as Felix said, this was a great opportunity to get in and actually prove, yeah, we can play nice together.

**Shane:** Okay, so let's talk about challenges. I can imagine there was a lot kicking off this project. Can you talk about what were some of the challenges you faced and how you overcame them?

**Jim**: I guess the first one was just scheduling. There were a lot of startup activities including training, preparations for the Quality Attribute Workshop on the architecture side. The sheer, you know, getting the right people at the right place at the right time was the first one. That's usually enough of a problem when you're just starting say a TSP initiative, trying to do all the startup on two initiatives was challenging at times. I'd say, Felix?

**Felix:** Yeah, that was quite. So get everything going, yeah. Also on there that the context that from the client side here, Bursatec, since they didn't really use a project like that, there was also not a clear idea of what the overall schedule would look like. So on one side that helped us a little bit because we had little bit more flexibility, but on the other side also there was no framework that we could lean on, so we really had to go through everything.

**Shane:** Right, and talking about the environment that you were in, you were not growing an e-commerce site, you're in the financial market growing a system that is extremely important, I would imagine. Did that...

**Jim:** Yes, this is the Mexican stock market. If you're going to trade securities in Mexico you're doing it here. I guess they've gotten over a million trades a day in terms of volume just since the beginning of this project they've gone over that benchmark, which was I think about the upper limit of what the original system was built to do. They're running at capacity right now, so you have that pressure added.

**Shane:** Felix can you tell us a little bit about the software engineering work and the role that ACE played in that?

**Felix:** First of all, just to make the context clear here, as Jim already said, there was not a lot of experience with the people on how to do this kind of work. That also meant in Bursatec they set up an architecture team that actually consists of four people tasked with a job to do all the architecture work for that system. From those four architects, there was actually only one who I would consider to be actually a senior architect, who we can trust who can do this kind of work. The other three who were developers were very good developers, but…junior. They didn't really have the training or the knowledge of how to tackle a system like this. That's why from the very beginning, we decided that instead of just teaching them we really need to coach them. To take care of that lack of knowledge, lack of skills to have them going through. What we also saw is that it was not just one of the ACE methods that they needed. They actually needed the whole package. So designing a system means that you have to have an understanding of your quality attribute scenarios. We have the quality attribute workshops to get those scenarios from the stakeholders. You have to know how to transform those quality attribute scenarios into a design. You have to know how to write down, how to document that design so you can actually communicate that to anyone who is interested in that. Of course you have to check, you have to test the design, if the design has been fulfilled. So they needed the whole package. They needed a set of all our methods combined in such a way that it helps in the environment where they are.

How we did that was that we started with a quality attribute workshop to get an idea of what the important quality attribute scenarios were. Of course, not surprisingly, the most important ones were performance and availability. You know, a system has to be available. So really high requirements here in terms of the amount of orders that go through the system. Of course, it has to be available while the stock market is open. So that was the initial one.

Then we set down with the architects and repeated a two-week cycle. In every two-week cycle we told them they take one of the scenarios. At the beginning of that cycle, we had a little brainstorming session to go through what are the probable solutions, how can you tackle that scenario, to have some ideas of what to go for. Then the team was sitting down for two weeks and coming up with a design. After two weeks we came back again and reviewed. At that review, which is using all the ATAM techniques that we have at the SEI, they had to really make the case why their proposed solution that they have will actually support the performance and the availability requirements. Going through those cycles every two weeks helped those young architects very quickly to learn what's important when you do an architecture design. They could focus. They could tackle the problem. They started thinking in terms of risks, so “When I do this, here are the risks that I have to take care of. How would I do that?” Going through that process very quickly, every two weeks, was a tremendous help. We also helped them also setting up the whole infrastructure, development infrastructure, the tool support, the processes, the documentation guidelines, and anything there.

**Jim:** Felix, there were a couple things there that you mentioned that I thought were the start of the happy circumstances for us. The first one is coaching. The whole paradigm of coaching really is central to the way TSP works. I'm a TSP coach. I don't think there was really that idea within the architecture group that coaching is a way to really accelerate the learning for a group of young developers and turn them into architects, but again it was very familiar to me. All I had to do was keep my hands off of the technical pieces and concentrate, as a good coach should be doing, on the team management aspect.

**Shane:** Is this happening at the same time? As the Quality Attribute Workshop is going on, is TSP being introduced?

**Jim:** Actually, the week of the Quality Attribute Workshop was the same week we were doing PSP training with the developers. Then, the next week, we launched the team, we did the initial launch of the team. That was an interesting experience just in terms of how much we were throwing at the organization in a very short period of time.

**Felix:** There was one interesting episode we had, it was about towards the end of the second iteration, which would be about 10 weeks into the architecture design, where we were almost at that point where we were ready to give up.

As you can imagine on one side, there is a deadline. There is time pressure. The system, the architecture has to be in a fairly mature state at a certain time. On the other side, we have those three young architects who need to learn and learning takes time. At the beginning, as you can imagine, there was not a lot of progress visible. I remember it was after Christmas here in Pittsburgh, so in Pittsburgh it was snowing, and the team was here.

In that time something amazing must have happened. All of a sudden, it was towards the end of the second iteration, every one of the team members started to get a feel of what it is that needed to be done, how to focus, and what kind of work to do. After that week that they stayed here in Pittsburgh, I was a happy camper again and they saw what can we do and, actually, all the time lag that we had, they were actually being able to recover. So they worked hard over Christmas, over New Year, didn't take a break, so they took care of all the lack of information missing and then we are on track going forward.

**Jim:** Felix I'm glad you never told me that you were near giving up. I knew that you weren't happy but from my point of view just looking at the data that the team was generating I could tell they were putting in the effort. I mean it was clear that they were going through and it was also clear to me that Felix wasn't happy but then we had, as he mentioned the Christmas break and you will see that a lot of times, especially with new teams, with younger teams, that they will go and they will beat their head against the wall and you'll think that they're never going to get through and then something happens or maybe something doesn't happen, they have some kind of a break and oh, now they get it. The light bulb goes on. So I've seen that happen before, I don't know that I've seen it quite in this way, of course you know I don't know if we've had this critical of a project in the hands of extremely junior developers for my stead, I think I did the calculation once the average age of the team overall of architects and developers was under 30. And I think it's now gone over 30, we're about 18 months into this. So, as you mentioned, a very young team, and with a lot of responsibility on them.

**Felix:** But again, just to add to that Jim, just for us, one of the lessons learned was that given the right environment, you know, being able to say you have the right infrastructure the right process, the right framework, the right goals for the team, actually also make a not-that-experienced team be successful.

**Jim:** That's—in my mind that's one of the advantages to having defined processes in place. And people that really had a little bit of training in how to use a defined process and keep track of the data, it actually gave them a little bit of feedback outside the architectural world, just, okay, we know we're working on the right things in the right order, okay. Even though the architecture coach might not be that happy we know we're sort of grinding through and that we'll get there. It—like I say, it's one of the rewarding parts of my job to be able to see this happen really in a lot of different environments, a lot of different people, different cultures even. And a wide range of applications of domains is the idea that process has value beyond, and if I really want to drive Felix and some of the architecture guys crazy I'll talk about, well, the architecture group has really elegant processes in place for doing this work. They don't necessarily want to call it that but I consider it a process for a very specific purpose, and that's architecture.

**Felix:** Yes, I mean experience shows that building, designing the architecture of a system to fulfill the goals is not something that you take a shower and you have the great idea, which happens every now and then so I don't want to say that you don't need that, but the fact is that probably 90 percent of the work is just hard work. You have days of processes you have to go through, you have to write down, you have to check and check and check and check … it's just work.

**Jim:** Yeah, the old idea 1 percent inspiration, 99 percent perspiration.

**Felix:** Yep. Therefore the architects may say 10 percent inspiration and 90 percent...

**Jim:** Fair enough, I'll take the 90's.

**Shane:** Just to clarify something you said earlier to anyone new to the SEI, you mentioned PSP; can you talk a little bit about what PSP is?

**Jim:** Yeah, PSP, Personal Software Process is really the training for Team Software Process. It's taking the ideas of defined processes and measuring that down to the individual level. And I think one thing that we've really demonstrated over the years is that software is very amenable to that approach. You know we've trained several thousand people in personal software processes. You know it gives you an effective feedback loop on yourself. Now it's not dependent on anyone else. Yes it can be helpful to have an instructor or a coach to help you along the way, but the idea behind PSP is you're installing your own measurement system, your own feedback mechanism so that you can learn and get better as time goes on.

**Shane:** Excellent. So we're moving now to the design of a system. Obviously there is lots of design at the top there that SEI has around and that's what we're going to talk about so we'll just talk a little bit about the work that went into designing the system. Where did you start?

**Felix:** Well, let me say one thing first. So over the years at the SEI… It’s probably now 15 or 20 years there, we learned that when it comes to designing a system, the one thing that is really important to understand are the quality attribute requirements. It is not that much the function that the system should have. The structure, the architecture of a system really depends on how well those functions can be done, so how fast, how easy it is to change them, how secure they are, how available it is, and so on. So that's the important piece. Because of this we also say you can only start in designing a system when you have an idea of what your important quality attribute requirements are. Which is one of the things that all the requirements as a notation method lack a little bit. Because the main goal is there to get a catalog of all the functions that the system has to have. Which is necessary and needed, but it's not the complete picture. The quality attributes come in there. And the second step is now since we acknowledged that designing a system means trying to achieve the quality attribute requirements, then of course you need to have a design method that does exactly this: focusing on the quality attributes and design in such a way that it actually fulfills those requirements. And I have to admit to my knowledge that although in the community it is acknowledged that designing a system means achieving the quality attributes, in most of the cases the method then after they acknowledge this, ignore the quality attributes and do something like function decomposition, which is probably the most popular method out there.

**Jim:** Or is if you're doing AGILE.

**Felix:** Yep. Those things. And they focus on the function of the quality attributes anymore. So to our knowledge, as Shane said, we have our own method, which is ADD, the Attribute Driven Design. That's the only method that I know that's actually taking the quality attribute requirements and transforming them into a design that will then actually fulfill those requirements. And if you have the structure, then of course the function requirements become important, you need to fill the structure with those functions. We also—in that example we talked about here in Bursatec—we can see a clear example of how that actually worked, because at that point when we started the architecture activities, the requirements phase was not finished. They had just—well of course they had some idea, the system has to do whatever the old system did…but all the new requirements, you know, we were just in the middle, of trying to find them out. But you couldn't wait because then, definitely, the deadline would be not achievable; so the only thing we had to do was have the quality attribute workshop to get an understanding of those requirements, and then let the other requirement elicitation move on and we could do the architecture design. There was only one requirement that came probably three months in which had a little bit of impact in the architecture. Not a lot so it took about a week to fix the architecture to accommodate that requirement. So we have clear evidence that that is true. You don't *have* to wait until you are done with the requirements.

**Jim:** And actually just watching what happened with the team because really the requirements changed throughout the project and today they're working on additional requirements that they accepted, they were actually supposed to finish it this week as we record this, but they've accepted additional requirements and it's really a matter of using the architecture to interpret the requirements and say, okay, this is where these are going to live within the architecture. And it's made it a very straightforward exercise for them to figure out both from a tactical point of view where do they live and how do we implement them, from a management point of view how long it's going to take. What resources do we have to bring to bear on this and when is it going to be done? So what I've been telling people lately is that this is one of the most boring larger projects that I've worked on from the point of view of, "We haven't had any major schedule crises." Now, part of that is the customer recognizing that this is a critically important project, they haven't messed with the resources, but even given that and especially given some of the other challenges that we've talked about, there haven't been any of those big management meetings where everything's in a shambles, we just never went there. You know, the architecture helped guide the project structure, the project structure helped keep things on track so when a requirement did come up that had some architectural impact there was no panic, there was no thought of anything other than, "Well, we're going to follow the process and everything will be fine." And it turns out it was fine.

**Shane:** And how long were these systems up? You have the new system, the old system, do they both run at the same time, or when do your take the old system down?

**Jim:** Okay, so that's one of the interesting things that's going to be happening over the next few months. So you know, as we sit here the project is actually very close to the end, this is literally the week that it was scheduled to finish, they were on track for that and they accepted additional requirements, they'll finish probably the middle of July now.

**Shane:** That was going to be my last question, when is this wrapping up?

**Jim:** The development phase should be formally finished, in fact I'm scheduled to make sort of a final closeout trip for the development stage in another couple of weeks here, you know, get the team's final data so we can do that analysis and put together a formal report, but basically the system is already working and has been for several months and they are, you know, implementing the last few functions that they need. They're now beginning to figure out, okay, how are we going to test with the other pieces of the system? We're working on the main trading engine but there are other functions that have to happen and work with the new system, as well as deal with some of the aspects of the older system that will still be in place.

**Shane:** Thank you guys very much for joining us today, taking the time.

**Felix:** It was a pleasure.

**Shane:** Thank you for joining us. This recording and a downloadable transcript are available at sei.cmu.edu/podcasts. If you have any questions please email us at info@sei.cmu.edu.