



<http://www.dotnetrocks.com>



Carl Franklin

Carl Franklin and Richard Campbell interview experts to bring you insights into .NET technology and the state of software development. More than just a dry interview show, we have fun! Original Music! Prizes! Check out what you've been missing!



Richard Campbell

Text Transcript of Show # 193
(Transcription services provided by [PWOP Productions](#))



Joel Semeniuk Updates Us on Team System August 17, 2006

Our Sponsors





Joel Semeniuk Updates Us on Team System August 17, 2006

Geoff Maciolek: The opinions and viewpoints expressed in .NET Rocks! are not necessarily those of its sponsors or of Microsoft Corporation, its partners or employees. .NET Rocks! is a production of Franklins.Net which is solely responsible for its content. Franklins.Net - 'Training Developers to Work Smarter.'

(Music)

Hey Rock heads! Stop soaking your beaver pelt in Molson brine and listen up, it's time for another stellar episode of .NET Rocks! Eh? 'The Internet Audio Talk Show for .NET developers' with Carl Franklin and Richard Campbell. This is Geoff Maciolek here to announce show #193 with guest Joel Semeniuk, recorded live Thursday, August 17th, 2006.

.NET Rocks is brought to you by Franklins.Net 'Training Developers to Work Smarter' and now offering a whole suite of onsite and remote classes in .NET 2.0 Technologies online at www.franklins.net and by telerik r.a.d. Controls, the most comprehensive suite of components for Windows Forms and ASP.NET web applications, online at www.telerik.com; support is also provided by Developer Express -- crafting first class tools, frameworks and controls for the .NET developer -- improve your experience online at www.devexpress.com -- and by CoDe Magazine, the leading independent magazine for .NET developers, online at www.code-magazine.com. And now, the man who, for the next hour, refuses to use the word Hoser, Carl Franklin.

(Music)

Carl Franklin: Thank you very much. You're listening to .NetRocks! This is Carl Franklin in New London, Connecticut. Of course I am talking to my partner out there in British Columbia, Richard Campbell. How are you Sir?

Richard Campbell: I am well; and is it my imagination or is 200 creeping up on us?

Carl Franklin: 200 is creeping up. We have something very special planned.

Richard Campbell: I am really looking forward to it.

Carl Franklin: Should be a lot of fun.

Richard Campbell: Yeah.

Carl Franklin: Reflective in nature I think.

Richard Campbell: Trying to do something different because the 100th show was so good.

Carl Franklin: It was, wasn't it?

Richard Campbell: Was a lot of fun.

Carl Franklin: Yeah. I am not saying that booze will be involved, but I am not saying it won't.

Richard Campbell: I am saying, by the end of that show, we are all going to be smashed. I am sure.

Carl Franklin: I don't know. It's still going to be good listening, I promise.

Richard Campbell: Okay.

Carl Franklin: It won't be Carl and Richard get shnockered on the radio; it won't be that.

Richard Campbell: No.

Carl Franklin: Richard, before we talk to Joel, I just got to tell the people about the .Net Rocks' TechEd Barcelona sweepstakes. If you have been paying attention in the last couple of weeks, we are giving away a ticket to TechEd developer in Barcelona, which you can read about at shrinkser.com/hhh. It's November 7th to 10th, which means I won't be at DevConnections in Las Vegas because of this, not because I don't like DevConnections just because of the scheduling conflict, but Mark Dunn is taking over for me there. We are going to be Podcasting TechEd Developer Europe from the show floor - interviews with people at the show. We are also going to do .Net Rocks! Live with a surprise guest, and we will be hosting a live presentation called 'Speaker Idol,' which I think has something to do with, people competing to get a speaker slot at TechEd. And all you have to do is, go to <http://www.dotnetrocks.com/barcelona> or shrinkster.com/hhi or click on the big Barcelona button. Now, there are two reasons we need to do this; one, we want to get a little more information about you, so that in particular how many downloads were actually -- how many people are actually listening to us, but also a little bit about your -- the line of work and all that kind of stuff so that we can better line up advertisers and keep the show on the air. But there is only a few questions, it's not a big deal. Once you register with us, you'll get one entry, per email address, per week - that means you are going to answer a question about the week's show and we are going to draw a winner from those people every week. Now, those aren't the winners for the contest, there is only one winner. But every week we'll pick a winner to get a choice of swag item from our 'Useless Crap Store' and on October 24th, we are going to pick a winner from all the weekly winners to go to Barcelona.



So, last week's question was, what two toppings did Steven Forte get on his Pizza at Piola? And the answer is, Onions and Garlic - and we randomly picked a winner from all the people who got the right answer. And this week's winner is, none other than Mark Engals from Rapid City South Dakota. Congratulations Mark! You get your pick of swag and you are in the running for the big "shoe." Now, this week's question is online at <http://www.netrocks.com/barcelona>. In the meantime, I also want to mention, we are going to be in Bulgaria -- October 9th and 10th at DevReach -- devreach.com, telerik is involved with that one, our friends over there. And we'll also be at the Tulsa TechFest, along with Mark Miller; he's going to be there in Tulsa Oklahoma tulsatechfest.com. And finally my last announcement is that, Joel Semeniuk is going to be teaching a five-day hands on, Team System class here at Franklin's Net New London Connecticut, the week of November 27th, right after Thanksgiving. You can get more information at www.franklins.net.

Okay. So let's talk to our guest, shall we? Joel Semeniuk is the founder of ImagiNET Resources Corp; a Canadian based Microsoft Gold Partner. He is also a Microsoft Regional Director and has a degree in Computer Science from the University of Manitoba. Joel has spent the last 12 years or so providing educational development and infrastructure consulting services to customers throughout North America. Joel specializes in helping organizations realize their potential through maturing their Software Development and Information Technology practices. Welcome Joel Semeniuk - welcome back!

Joel Semeniuk: Hey! Thanks a lot guys, good to be back.

Carl Franklin: When -- you know, you basically introduced our listeners, or at least us to the best explanation of Team System we'd ever heard.

Richard Campbell: Absolutely.

Joel Semeniuk: That's great. Good news.

Carl Franklin: So, Team System has been coursing through your veins in the last couple of years, I take it.

Joel Semeniuk: I live, breathe and eat Team System; there is no denying that.

Carl Franklin: How does it taste?

Richard Campbell: But I realized, when I was doing show planning that the last time we talked to you about Team System was before

Joel Semeniuk Updates Us on Team System August 17, 2006

foundation server had even shipped. So, we really are missing out on some big pieces of this stuff and it's getting mature now, isn't it?

Joel Semeniuk: It is starting to settle down, you bet -- I mean, obviously at TechEd we've seen the new release of Database Professional for Team System, which has really added a sweet spot to the value of Team Suite.

Richard Campbell: Oh, you are speaking my language; I am the database guy, I am really excited about data -- 'data dude' as it was once called.

Carl Franklin: Yeah, we love that.

Joel Semeniuk: Amen brother! We love that. And team foundation is really kind of a -- you know, as its name suggests, the foundation for more goodness and that was released early this year.

Carl Franklin: More wholesome goodness!

Richard Campbell: More wholesome goodness; and I am kind of surprised, you know you talk about -- that's the foundation, how could they ship the product without it?

Joel Semeniuk: Well, I mean, to tell you quite justly, you know, the Team Suite aspect has really much more of an extension to the existing developer tools that were already out there. So, we had just some amazing functionality around unit testing and doing design work. Team foundation is really there for the connection of all the stuff - the collaboration component - really being involved in things like Bug tracking and so forth. So, it's really the collaborative heart, the data warehouse heart of the product that extends not just the tools itself, but how those tools interact with the team.

Carl Franklin: So Joel, clear this up for us, does -- is Team System Foundation Services required in order to do work items and bug tracking and all of that stuff?

Joel Semeniuk: Yeah. That's really one of the core services that Team Foundation Server provides -- is in fact Work Item Tracking. So, in a nutshell, Work Item Tracking -- Team Foundation Server just allows you to track stuff. Work items are declarative pieces of stuff that you want to track, and one of them happens to be a bug and there are other things like tasks and issues and risks and so forth.

Richard Campbell: It's good -- I am glad they are calling them work items because you know, I have worked with a lot of different companies



Joel Semeniuk Updates Us on Team System August 17, 2006

using various Bug trackers, and they use them for way more than bugs -- where we put in our new feature ideas. I have even gone so far as to write bugs on how to do analysis on a particular chunk of data; so that we're really sort of accumulating a knowledge base of how this particular system works? So, calling them bugs is really undermining the true significance of what a tool like that does.

Joel Semeniuk: Yeah, I mean, you can't really say it's a bug-tracking system, yet it is an amazing bug-tracking system. You know, and what's even more powerful is that it's totally extensible. So, if you wanted to create your own Work Item Types or you wanted to change or modified existing Work Item Types, let's say to add fields or to take away fields for example, you can totally do that, which allows you to track so many different things like, for example, on my team, we typically track personas, which are kind of like actors on a used case inside of Team System; that's just another thing that we want to track. That's not built into Team System, but it was -- takes us about 10 minutes to extend it and allow us to have a list of personas.

Richard Campbell: I hear this over and over again about Team System that everything is extensible; that they are not making any final definitions of how you do anything.

Carl Franklin: Right. So, give us an example of what you could -- how you could extend it? How you could modify the behavior?

Joel Semeniuk: You know, this is really -- the thing that really gives me goose bumps -- in fact, I am actually standing and walking around my office right now, and just talking about the extensibility story has really been some of our successes over the last year. It hasn't necessarily been the features that have been out of the box; however, they've been really good, but it's the ability to extend them and modify them to our needs. So, the first place that most customers actually start looking at when it comes to extensibility is in fact Work Items. So, Work Items are stored and defined by XML files. So, if I wanted to go and modify Work Items, I can actually use some tools to kind of extract the definition out of Team System into an XML definition. I go and modify that XML definition; for example, I can add fields or add rules like workflow inside of the Work Item definition itself and then re-import that XML back into the server and lo and behold, we have got a new Work Item Type definition with some new rules that will govern how it's used. So, a lot of the people actually change, for example, like mandatory fields; they might want to ensure that the triage -- the field is filled out in a certain way for example,

or that a particular rule in workflow is followed. So, for example, I can't add a new requirement until it's been approved by someone.

Richard Campbell: That's nice because it's how you are starting to enforce your own rules. I mean, we are so adamant against Microsoft forcing us down a path. But when you are actually governing a team; you want to have enforceable rules like stuff like, it must be triaged, it must be approved those sorts of things, to be able to set those guidelines yourself.

Joel Semeniuk: Or less rules -- I mean, I have worked for organizations that have the book, you walk into the organization and say, "This is how we develop software; this is our software development methodology and you go, "But I am only here for three weeks and I have got a team of myself. Really? Do I have to follow all of these rules?" The reality is, is that when I start working with customers and we look at their processes, and as a whole, we found out there is really kind of clusters of rules, based on the types of projects people are working on; you are working on a really large application with business analysts and testing departments and so forth - those are going to be different rules, different methodologies than an agile team of three guys is trying to figure out a little workflow application internal to the organization. So, let's embrace that. Let's embrace those differences and have different rules for each type of the project that we are working on.

Richard Campbell: You know, it's an interesting problem we are working on right now, my project is well in the development cycle; and it's not a big team. It's a small team of people that's building software. But when you are talking about networking, the stuff must be a 100%. So, we don't have a lot of code and we don't have a lot of people, we have a lot of process. We have an unbelievable amount of testing; the number of test cases, every time we make any change, we got to check every test case to make sure everything behaves still. And I just can't imagine building this kind of software without the kind of framework that we have with the modern development tools.

Joel Semeniuk: Agreed. And Work Items play a huge role in that because they allow you to track stuff. So, I've worked in many organizations that have tracked stuff in documents. And quite frankly, documents aren't the best places to track stuff like requirements, like features, like bugs, like tests and so forth. What you want is to be able to track those pieces individually from each other, but also have good association traceability between all these different things, so that if I have a requirement defined in my system, I might want



to have that requirement traced to all the test cases that validate that that requirement is still met in the software. So, if my requirement changes because of my traceability, I should be able to find what test cases I have to actually go and modify to ensure that my new requirement, or the new version of the requirement is still satisfied in the software.

Carl Franklin: Yeah.

Joel Semeniuk: That's hard to do in docs.

Carl Franklin: Seriously

Richard Campbell: That is really hard to do. And I could see because of the integration with Studio, the capturing things like who worked on it and for how long they worked, and what their work actually was sort of, falls naturally out of the tool.

Joel Semeniuk: Right. Well, what's really cool and has really added a lot of value underpinning the application -- the Team Foundation Server, is the Data Warehouse. So, we have these work items -- and it gives us nice list, and we can track things and they go to workflow and buildages are generally dancing because of it. But more importantly all of the data, all of the state, and all of the history goes inside of some big large data warehouse so I can understand trends;, I can understand the relationship between different types of Work Items and even different Work Items and quality for my team itself. I can understand how my project is progressing, I can interpolate some future impacts that we might not be aware of, if we are just looking at the data and the lists themselves -- and that is really quite powerful.

Richard Campbell: Now, are we interpolating or are we extrapolating?

Joel Semeniuk: Extrapolating is probably the better word.

Richard Campbell: But the idea being -- of course, one of the major questions the business guy is going to ask and hey, let's face it, with the strange loop side of things, and the business guy is, 'When will it be done?'

Joel Semeniuk: Or more importantly, will it be done on this date?

Richard Campbell: Right.

Carl Franklin: Right.

Joel Semeniuk: You know we said it was...

Joel Semeniuk Updates Us on Team System August 17, 2006

Richard Campbell: In fact, it should of course be 'No.'

Joel Semeniuk: Yeah, exactly. That's the standard answer.

Carl Franklin: So that's the thing that it's very easy to get a instant snapshot of your project state at anytime rather than having to set up and run a special report; it's just there right?

Joel Semeniuk: Yeah. Well, there's this really -- one really great report and I guess this leads us into the -- we are kind of, jumping all over the place but there is one really great report that I use at a number of my customer sites, it's called the Remaining Work Report and it's a cumulative flow report which kind of shows you the trends of how things are going over time. And by looking at this report, you can actually see Trendlines are going to be resulting from it and you can basically determine whether or not, based on your current velocity, your current quality metrics, will you actually hit some deadline; will you accomplish what you said you were going to accomplish at the start of your iteration or project.

Carl Franklin: Has it been accurate?

Joel Semeniuk: Well, there's always fluctuations in a project but it does give you a good feel. So, it's not meant to be a very precise measuring tool, most are going to give you feedback around general trend.

Carl Franklin: Most reports are like that anyway.

Richard Campbell: I am wondering, does it gradually get more accurate over time? That it's accumulating knowledge about the way the developers work and sort of get a sense -- you learn more about how long things actually take.

Joel Semeniuk: Well, that's an interesting question because -- and a number of customers, we look at these reports and we say, "What can we learn from these reports?" And it turns out, -- "Well you know, if we entered work items in this way and track them a little differently we'd actually get much better reports." And so, it allows you to kind of fine-tune your tracking system depending on how accurate or what type of 'Ouch' you want from your report.

Richard Campbell: Right.

Carl Franklin: Yeah... good feedback.

Joel Semeniuk: For example, on a lot of projects that we started off, when do we go back and update Work Item? So far I've got 50 bugs assigned to me and I've knocked those bugs off



during my development cycle; do I at the end of the week go in into my reporting -- or sort of, my Work Item tracking and close off those 50 bugs or do I close them off as I go forward? So, interesting question, it seems like a no-brainer but interestingly enough, how you do it would affect what that little chart looks like.

Richard Campbell: Right.

Carl Franklin: Sure.

Joel Semeniuk: You're going to get lots of bigger steps inside of your charts that show cumulative flow of bug closing versus if you have to close bugs off everyday and you get a more natural, kind of, real time trend of how things are flowing.

Richard Campbell: More of a curve.

Carl Franklin: Awesome!

Joel Semeniuk: Yeah, exactly. So, little things like that have forced us to -- we'd like see trending everyday instead of at the end of an iteration or at the end of a phase of a project for example. So we try to interact with the tool daily. So as we're working with our bugs we're closing bugs off; as we are delivering features into production, we are marking out that state in that way as we're doing so. And we actually get more -- I won't call them 'Realtime', but say more 'Realtime' trends of what's happening on my project instead of seeing the results... Like, a traditional project at the end of a phase. So, how did we do guys, do we meet our timelines are we going to meet the project? You want to have more immediate feedback on that.

Richard Campbell: I can see that the reports close the loop; that finally most developers use a bug tracker highly resistantly - the minimum effort possible; that's why you end up finishing up all your bugs at the end of the week because you want to -- that's when the reporting period is. But if you were posting that graph everyday so that it shows, people get into the routine of, "Make sure you update everything everyday," because you can see it everyday. It's a simple feedback mechanism; if you're not going to get feedback, you're not going to change behavior.

Richard Campbell: Right.

Joel Semeniuk: So, we need to have something that's going to give us feedback without a project manager having to do all that logistical stack stuff themselves...

Richard Campbell: Right.

Joel Semeniuk Updates Us on Team System August 17, 2006

Joel Semeniuk: ...which is traditionally -- if we really wanted to have that -- those good metrics, it's up to the project manager to set up processes to first gather those metrics, and then second of all, enforce that developers conform to processes that generate those metrics. And then the consolidation of that data into something meaningful back to the developers takes time. This is just done automatically; which is awesome.

Carl Franklin: Yeah.

Richard Campbell: Yeah, low labor. Just happens all -- that's how you make it reliable, that it takes very little effort for it to always be there.

Joel Semeniuk: You know I am writing a book right now called '*Managing Projects with Team System*' and it's all about thinking about the management of projects to make them successful not necessarily the logistical side of project management. Ultimately, we do logistics in order to give us metrics and a way of baselining our processes so that we can get better. But wouldn't it be great if we didn't have to do the logistics so that we can understand the metrics and allow our project team to be more proficient and effectively higher quality deliverables, and focus more on the management side, not the tracking and logistical side of what we're doing.

Richard Campbell: Yeah.

Carl Franklin: Joel, you seem pretty bullish on Team System overall; do you have any complaints or anything you'd like to see work a little better?

Joel Semeniuk: Well, it's a V1 product, and with that there are going to be some initial problems I guess you can say. We have in this -- and a couple of different customers kind of uncovered some load issues. And Microsoft has been very, very open about how they have been addressing scalability. Brian Harry's blog -- he goes into a great deal of detail into how they have addressed the scalability issues internally at Microsoft, but we've been seeing them as well at customer site. When you get into a large amount of files, and if you get a large number of Work Items you will see degradation in both the data warehouse and certain operations when it comes to Version Control. So, obviously those need to get fixed.

Carl Franklin: Do you think that's something that can be fixed at the database with indexing and...?

Joel Semeniuk: Well, they are tweaking it all over the place. For example, there's some good stories out there that says, you know, we tweak



this one store procedure and we got 400% more efficiency over this particular operation. I've heard from the product team that they are looking to go from operations that take 10 minutes to 15-20 seconds; it's just by changing, kind of, how they are working with the data under the hood. They've uncovered a lot of locking issues at the SQL Server level for example; well, let's address those.

Carl Franklin: Because I am thinking it's time to call Kim Tripp, right?

Richard Campbell: (Laughter).

Joel Semeniuk: Absolutely. She should be on the advisory committee for that one for sure.

Carl Franklin: (Laughter) Five minutes, "Ah! Here's your problem - tweak - gone, problem gone."

Joel Semeniuk: Yeah.

Richard Campbell: I am looking at Brian Harry's blog; just talking about his stats on them dog fooding their Team System themselves and they have got a SAN to run this thing that's got 15-paired 300 gigabyte drives in it and a spare. Now, that's a big database.

Carl Franklin: Yeah.

Joel Semeniuk: Yeah, it's nice.

Richard Campbell: But look at the -- you mean, the stats are crazy; 750 users, 200,000 Open Work Items.

Carl Franklin: Richard, where are you reading this?

Richard Campbell: This is off of Brian Harry's blog, just talking about the July stats...

Carl Franklin: Well, I have a link to that.

Richard Campbell: ...of them consuming their own dog food using Team System in-house. And there -- they are beating things up.

Joel Semeniuk: And their goal is scalability; I mean, in a lot of cases most of my customers aren't hitting these thresholds...

Richard Campbell: Now, it makes sense to me that Microsoft has got huge projects and I hope that they don't emphasize that aspect of Team System so much because there are so many people out there with their 10-15 person shops that really need this stuff.

Joel Semeniuk Updates Us on Team System August 17, 2006

Joel Semeniuk: Oh, yeah.

Richard Campbell: We really need a solution at that scale.

Joel Semeniuk: Absolutely. And at a departmental level as well, I mean, I am working with a number of large organizations that is really doing this at the enterprise scale and they are really doing it departmentally, and they are having a lot of really great successes with the tool. They are adopting it, their methods of adopting are -- they are kind of doing the no-brainers first. It's kind of like, well, we have been using Visual SourceSafe for the longest time and we want out --

(Laughter)

You know, someone say this. And then, out comes Team Foundation Server and for just a brand new Version Control System alone, they believed that the mitigation and risk is justifying the purchase of the product. And then we get all this other good stuff that they will learn to use eventually like, bug tracking, like issue tracking, and crazy enough, things like requirements, you know, ooh!

Carl Franklin: Yeah, right.

Richard Campbell: I am thinking, again as a business owner, being able to track the true cost of a feature; like, to go through and say, "This is when we put in the bug, this is how long we worked on it, this is how much it really cost us in effort and time."

Joel Semeniuk: Well, see, that's where I would say that Team System probably doesn't do a great job -- and I will tell you why. A lot of companies that I've worked with have tried to take Team System and turn it into a time tracking system. Please record Mr. Developer, Mrs. -- whatever Developer, how much time you spent on that bug, on this task, on this feature? And it really breaks down. It's really not a good tool for time tracking, therefore, if it's not tracking time accurately, it's very hard to get overall, a bigger picture of total cost of ownership or say total cost of development for a particular feature for example.

Carl Franklin: Well, now it's extensible; is it enough to write a plug-in at the client that can interact with Team Foundation Server?

Joel Semeniuk: Yeah. There is a couple of companies, one of them is Avanade who is writing a plug-in with Microsoft Project Server that will allow you to better track your time inside of Project Server and take some of that metrics and



push them back and forth between Team Server and Project Server, that tool -- to my knowledge hasn't been released yet. And I know that Microsoft has been getting a lot of feedback on this particular problem in the lack -- the general lack of integration between Project Server and Team Foundation Server that would allow for these types of metrics to be harvested properly.

Carl Franklin: Well, Joel I've got more questions for you, but before that I want to take a minute to say that this portion of .NET Rocks! is brought to you by our good friends at Developer Express - Developer Express, 'Crafting first class tools, frameworks and controls for the .NET developer,' - improve your experience online at www.devexpress.com.

(Music)

Carl Franklin: Joel, you mentioned a couple of bugs or if you will, issues, but what about features that you wish were there? Then the V2 features, whether Microsoft is thinking about them or not, I am sure they are, but what would you like to see?

Joel Semeniuk: Personally, I would like to have a better design and integrated experience for the Project Manager. One of my biggest pet peeves is in iteration. And iteration in the project management world is a block of time that we are going to use as a bucket to do stuff in. Now, an iteration inside of Team System acts more like a simple label; it has a name that we can assign to things -- to Work Items. It doesn't have any context, it doesn't have any date ranges, it doesn't have any constraints around it. I would like to have or to see the iteration become more real inside of Team System, so that as a Project Manager, I can naturally visualize the flow of my project through that really important concept called an 'Iteration', and right now that's generally lacking. I'd like to see hierarchical Work Items; a lot of people break down problems hierarchically, and right now there is no concept of hierarchy inside of the Work Items...

Carl Franklin: So you can't really link one to another one? Is that what you are saying?

Joel Semeniuk: Well, linking is there, but it's non-directional. So, in other tools that you see on the market, you have the concept of a 'Trace From' and a 'Trace To'. So, for example, I can have a requirement and then task and implement that requirement, 'Trace From' that requirement. Now, inside of Team System, linking is non-directional; it's just as I am linking to those Work Items and there it is.

Joel Semeniuk Updates Us on Team System August 17, 2006

Richard Campbell: Thinking about the way that I worked on projects using other products where, often we will build sort of the parent bug where the parent Work Items is, this is the business case for the problem and we know it involves these Servers, Developers, and these UI developers and these database developers. And we'll create separate subordinate tasks for each one of those different pieces, and it knows my VB composition the Work Items for individual people.

Joel Semeniuk: Yeah, right.

Richard Campbell: And you don't finish the top one until all the lower ones are discovered and of course, one of the things you find out as soon as you dig into that like that is that there are other issues you didn't think of and so the thing cascades quite a ways.

Joel Semeniuk: Yeah. Well you've just actually hit on something that is also really important, and that's 'Cross Work Item Workflow', which is kind of what we are talking about, we are saying, you know...

Richard Campbell: Like a hierarchy in a dependency tree.

Joel Semeniuk: But more importantly, like if a requirement goes from 'Proposed' to 'Accepted' stage, wouldn't it be great if five other work items are generated saying, okay, now I've got to do these things to implement that requirement. Right now, there is no workflow that spans Work Items; meaning that I can have workflow within -- it's really just a state graph. So, I can really have a state graph inside of a work item such that it has to go through particular states, but those states do not interact with the other states of -- or states of other work item types like bugs or so forth. You know for example, wouldn't it be cool if I changed a requirement definition and it went from completed to -- in progress again, for example, which means I am changing the requirements. Wouldn't it be good, if it decided to go through all of the tasks and the bugs and the test cases that were related to that requirement and flag them as, "Ooh! You might have to look at the tasks that are associated to the requirement that I am changing," for example. To have that level of extrapolation isn't quite there; you have to kind of dig into, create special queries and dig into the warehousing sometime.

Carl Franklin: Now Joel is this something that you can -- you can compensate for by extensibility?

Joel Semeniuk: Well, see there's the good story. The extensibility is really the key there. I've been doing probably more extensibility work than



anything else lately. I mean, we were the first to create a Process Template Editor, which allowed -- I mentioned before that work items are defined inside of XML, but really the entire project structure, the Work Item types, the queries that you see, the documents that are inside of the document library and so forth, they're all defined inside something called a 'Team Project Template'. And that's just a whole cluster of XML files. Now, XML is wonderful, except when you have referential integrity between the XML files and so forth. So, what we did is, we built a GUI on top of this XML that allowed companies to much more easily modify the definitions of Work Items and the Process Templates themselves, and publish them back up to Team System. More importantly, our tool also, with the extensive set of APIs, further validates that XML before it actually publishes that to the server to help cut down on some of the round trips if you'd made some boobos. So, the extensibility is profound; the APIs around Version Control for example, are huge; there is lot of stuff that we can do in Version Control out of the box, but if you look at the APIs, the sky is the limit. I had a guy in my class in the UK about a year ago, who was talking to me about false *showblame* function, where, you know, I want to see the person who touched that line of code last, but I can go and bonk him on the head.

[Laughter]

And the reality is, that out of the box, we don't have a show blame feature. But using the APIs -- and I am not kidding you, within four hours, he created a utility that showed blame.

Carl Franklin: Now, is this done with a plug-in interface or interfaces or, you know, the plug-in architecture idea, components, Raw Objects and APIs, what's the implementation syntax like?

Joel Semeniuk: It's pretty broad. Most of the customization is done through client side APIs, which are very, very robust. So different -- there is a Team Foundation set of objects that you can call -- you make references to in your code and just get to do stuff for you. There is also a pretty extensive plug-in model, which means that you can develop your own services that plug right into Team System and extend some of the services that are there. However, that's really more for the vendor, the ISV's that are out there, that are looking to literally extend the core products functionality. So, most people who are doing extensibility are just using the published API, they are normally found on the client side.

Carl Franklin: And that brings us to the next question, which is, what is the third party market like? Is it growing, does it exist?

Joel Semeniuk: Well, there are lots of people who are plugging in and integrating with Team System. I've seen a few, for example, some requirements -- requirement management tools that have started to spring up, that integrate with Team System. I have seen some extensibility in some of the designers. So, while you are working with -- for example, the application modeler inside of Team System, some of these vendors have actually extended the modeling service to provide more functionality to the developer -- to the architect while they are using it. So, it is a growing market and I think one of the frustrations that I have heard from a lot of the vendors that I know who are developing products for the company is that, the tool is still young - it is still kind of early. The APIs are robust but you know, what's next, what's coming down the pipe? How are they going to leverage them?

Richard Campbell: Yeah, I was thinking that. What's going to happen in V2? How badly are these things going to break? I think about VBX to OCX and how badly we got hammered? People who are early into this thing are going to run into a lot of issues because Microsoft is learning a ton and they are going to make substantial changes.

Joel Semeniuk: Yeah, well one of the things that we are looking at right now, is a new feature that should be released in the service pack, which will be around the end of September or so. And one of the new features is to have a custom control set up a Work Item. So, by default, you have just the set of controls that Microsoft provides you as part of a Work Item, which are pretty much drop down lists and text boxes and so forth. But what if you wanted to put a grid in there, what if you want to have your own control that - fax information from a different store...

Carl Franklin: Hell yeah!

Joel Semeniuk: and presents it differently. And that's a great idea, but I wonder, well, what's that going to do to the process template editor that we created? And what's that going to do to deployment? How am I going to deploy those CX's or whatever you are using, DLLs to my client side computers. We are working on...

Carl Franklin: No, I am sorry that's too much power. You can't have that feature. (Laughter)

Joel Semeniuk: Yeah, turn that off. We are working on a requirements management tool for Word and Team System. So what it does is that round trips -- I can actually start typing out a use case inside of Word and then have that use case published to Team System and then have it round tripped back into Word. And more



importantly I can have – we created something called the Work Item Explorer that I can drag Work Items from this Explorer to my Word service and it renders that according to a particular template that helped in document generation. And you can also pull up the Work Item Editor right in -- right inside of Word to start filling out some of the details and we are all thinking well, geez! How is that going to impact our app? So, we are kind of in a wait and see mode as well, to see what the APIs are going to look like, how the XML definition that actually defines the Work Item is going to have to change and so forth.

Richard Campbell: Joel, poking around on the third party side of things, I saw this mentioned on your blog, “Scrum for Team System.” And I am a Scrum junky, so this is pretty exciting for me.

Joel Semeniuk: Yeah, I am a Scrum junky myself. What this really is, is a customized process template. I mentioned to you before that Team System runs off of these process templates that define all these Work Items and so forth. So what ‘Scrum for Team System’ is, is really a customized process template that has their own Work Item definitions, and we have things like Sprints and Backlogs and so forth.

Carl Franklin: Right.

Joel Semeniuk: And it also comes with -- when you install it, it also installs a Scrum informational site that tells about -- tells the user how to use the methodology, what all these concepts are and how to manage the project to the reports that it gives you as well. So, Scrum comes with new Work Items, a new project portal as well as brand new reports that ScrumMaster or the Team itself might want to use to see how they are doing, and how to schedule the next iteration and so forth – sorry, the next Sprint. It is really well written; it is really quite a powerful methodology and I recommend that to the world to see because it is definitely a viable methodology.

Richard Campbell: When Team System first came out, I remember us talking about this, the fact that people were going to be able to build templates based on different development methodologies, and actually we were talking about XP, as Scrum is a flavor of XP, and I think what’s exciting to me about this is this idea that if you are a Scrum guy who doesn’t know Team System, this will make Team System more familiar to you; but the other way around works too, you are a Studio guy who doesn’t really know how to do Scrum, this will help you give some structure to how you use Scrum effectively.

Joel Semeniuk Updates Us on Team System August 17, 2006

Joel Semeniuk: Absolutely. And I have to applaud Microsoft for opening up that ability. It would be sad if Team System only supported MSF...

Carl Franklin: Right.

Joel Semeniuk: ...for example. They have really kind of taken a different approach and said, you know what, we are shaping MSF as methodologies, and they are good. But we are just -- we are not closing the door; if others out there want to have one for the unified process for example, sure, go ahead, go nuts.

Richard Campbell: Do it the way you want to do it.

Carl Franklin: Well, ultimately I think Microsoft’s goal is to provide the best tools for software development, whatever direction software development goes in, and they do have a good idea of software development.

Richard Campbell: You’d have to hope anyway.

Carl Franklin: You would hope - yeah.

Richard Campbell: Is there any greater compliment for a programmer really than somebody using your software in a way you didn’t think of?

Carl Franklin: Right.

Richard Campbell: You know, it’s really a testament to how good your software is, that even though you didn’t think of this use case and build it in, somebody was still able to take your tool and use it that way.

Joel Semeniuk: Yeah. Absolutely. And you go, “Cool.”

Richard Campbell: We sort of bammed against this a little bit; the deployment characteristics of Team System. I don’t think TFS really has much to do with that, does it?

Joel Semeniuk: Deploying your application?

Richard Campbell: Yeah.

Joel Semeniuk: No. It really doesn’t come into play with any type of deployment scenario.

Richard Campbell: Other than when you do the Build, it’s still a part of the check-in process and making sure everything is properly packed away, marking it as a Build during the steps involved.



Joel Semeniuk Updates Us on Team System August 17, 2006

Joel Semeniuk: True, I mean the Build is part of that; I don't personally consider Build as a deployment -- as an aspect of deployment. However, we have been using the Automated Builds to aid with deployment. So, we would actually go and create a release to production Build, for example that -- instead of having to run through all the unit tests and all the checks and all that kind of stuff, it's really just kind of there to recompile the application in release mode and dump it to a particular location on our network, which we can then go and pick it up and deploy to our web servers or something like that.

Richard Campbell: Right. That makes sense to me.

Joel Semeniuk: Yeah.

Carl Franklin: And now, hey guys before we -- before we get back to it, I want to mention that this portion of .Net Rocks! is brought to you by our friends at telerik; telerik r.a.d controls, r-a-d controls is the most comprehensive suite of components for Windows Forms and ASP.NET applications. These guys do miracles with AJAX. Check them out online at www.telerik.com.

(Music)

Richard Campbell: Joel, this walks nicely into another topic that's a favorite of mine, which is Continuous Integration.

Joel Semeniuk: Absolutely.

Richard Campbell: And Microsoft did not really build the concept of Continuous Integration into Team System; and I guess they really shouldn't have, so they didn't, because it is one particular methodology to building software.

Joel Semeniuk: Right, and you know, I don't think it was -- we are not going to support Continuous Integration type of conversations. It's kind of like, we have to ship a product, and what can we get away with not shipping in V1, is the Continuous Integration metaphor important to Microsoft? Absolutely. In fact, all the bits and pieces are there for Continuous Integration. The APIs are rich enough; we have implemented Continuous Integration at customer sites that leverage these APIs. So it's totally possible with the tool; is it built in and right out of the box? -- Unfortunately not.

Carl Franklin: Well, you know. Visual Studio Team System runs the risk of becoming the word of development, right?

Richard Campbell: Right.

Carl Franklin: If they start getting feature-creep and featureitis, pretty soon they'll be asking the question, well what is this, right? So, I mean there are other tools that -- do Continuous Integration that -- that's what they do.

Richard Campbell: And I am thinking of Cruise Control, of course

Carl Franklin: Cruise Control of course, yeah.

Joel Semeniuk: Yeah. Well, Cruise Control does have the ability to integrate with Team System. It's probably not a 100% unified Team System based Continual Integration solution but it -- it can use Team System to help control or provide the source and provide the control for the continuation -- Continual Integration base. It works pretty slick.

Carl Franklin: Okay.

Joel Semeniuk: For those people who are actually already using Cruise Control against Visual SourceSafe for example, they shouldn't fear moving towards Cruise Control against that and using Team System. There's a natural mindset that is still there, but of course we are probably going to be seeing much more Continual Integration tools from Microsoft as well as from other vendors as well.

Carl Franklin: Joel, give us your favorite feature, I mean we've talked about a lot of features; what's the coolest bit of this?

Joel Semeniuk: Absolutely the API. I mean, each of the features themselves are -- some of them are cool, some of them are maybe not so cool; like the Microsoft Project Integration, I would put on the not-so-cool side. However the APIs, I just -- we have no end in fun in what we can do with those APIs; I have a summer student working with me, ironically his name is Joel. (Laughter) Immediately I labeled him as a genius, (laughter) but he is doing some phenomenal stuff with us, around Vista, Office 2007, and the Team System APIs, and he's just wading into the stuff, consuming it. He is not lost in these APIs, are really well written, they work and we're just doing some extraordinary stuff with that. And I just find this pattern over and over again; if it's not there, it's not really that difficult to go and make it do that stuff -- and you shouldn't be intimidated by it. So, I really get jazzed when it comes to the extensibility side of Team System, from the server perspective.

Carl Franklin: And again, you think it's kind of too early to point to a really robust Third Party Tool community. But you think that's going to happen?



Joel Semeniuk: Yeah. It's going to happen for sure. But I don't feel the love quite yet with regards to an entire segment of the IT community that's building add-ons for Team System yet. It's going to happen though.

Richard Campbell: You know, we talked a bit about XP being used inside of Team System. Have we seen a full CMMI implementation under Team System as well?

Joel Semeniuk: What do you mean by full, Level 3?

Richard Campbell: Well, okay, 3 is enough if you can get a Level 3 implementation. And I guess the main thing here is, are we following the reporting methodologies dictated in CMMI so that we got that feedback mechanism that they want?

Joel Semeniuk: Yeah. CMMI is the process template that Microsoft ships -- it's called MSF for CMMI. And at the time of launch, I have to -- I don't want to get the Microsoft guys mad at me, but I have to say it was partially incomplete in some areas. The Work Items were defined improperly. I didn't think that there was enough guidance in the process methodology to instruct consumers of that methodology and how to actually implement MSF for CMMI to achieve a CMMI baseline. That being said, I don't have any customers who are using Team System and the CMMI capabilities to specifically achieve CMMI. What I've been finding is that a number of customers are saying, "Hey, we kind of like CMMI; we are not interested in getting certified but we like some of its principles. Let's use MSF for CMMI and the good tracking and auditing stuff that Team System provides for us, to increase our practices and to help us make a more predictable process model." That's the general trend. Personally I haven't seen a customer who is trying to achieve MSF for CMMI Level 3 with Team System yet. That doesn't mean that one doesn't exist and I'm sure the Microsoft guys are out there, please go and blog about someone who's done that and that would be really a great story.

Richard Campbell: So, we talked about -- have we covered all the features of Team Foundation Server yet, or we have been certainly talking about the reporting side of things. We have talked about the Work Item, that whole workflow component. Are there pieces we have missed?

Joel Semeniuk: Well, yeah I mean, it all kind of, starts with Work Items. The Version Control is obviously brand new, it's not built on top of Visual SourceSafe, and anyway, it's a brand new Source Control Engine.

Carl Franklin: Yeah. Bye-Bye VSS. So long.

Joel Semeniuk: You know, underneath the hood, we have the data warehouse and it's interesting how Team System Foundation has actually been implemented in a pluggable architecture. So all these things, like Change Management, the Work Item Tracking, they are all just plug-ins for Team System and they all have their own -- what are called Operational Stores. So, Work Items; Plug-in for example, saves its data to the Work Item Operational Store. The Automated Build piece for example, saves its data to the Build Automated Store. And then, what happens is that, Team System is able to aggregate data from all of these desperate Operational Stores, into one data warehouse and allow people to report on that. And really that's -- it's kind of an untouched feature, but it's probably in my opinion, one of the most valuable ones for producing that feedback mechanism. And then, of course, on top of that are all the reports. And there is static reports, which are the reports that we get out of the box, which are -- some of them, really great and some of them -- you kind of wonder, "Hey, why do I have that report?" (Laughter) But then you also have the Microsoft Excel Integration. So, I mean it's just a cube. So I go into Excel, I go open a new data source, I point to the cube, I suck in the data and then make all my fancy-dancy pivot table for that. That is good stuff.

Richard Campbell: Yeah, that'll keep a manager busy for hours, so it doesn't bother anybody.

Joel Semeniuk: Yeah, just don't play with your queue.

(Laughter)

Richard Campbell: Don't play with your queue. We'll talk to you later.

Joel Semeniuk: Yeah.

Carl Franklin: Oh! We are such smug developers. We should be slapped.

Richard Campbell: I'm making fun of myself, I swear.

Joel Semeniuk: I know. Yeah, here I am writing a book on Project Management, right?

Richard Campbell: That's right, yeah.

Carl Franklin: So, what's coming down the pike, in terms of Team System?



Joel Semeniuk Updates Us on Team System August 17, 2006

Joel Semeniuk: Well, we are going to have a new Service Pack. New Service Pack is probably going to be the biggest news that we are going to see this year, coming up in September.

Richard Campbell: This is not the first Service Pack, is it?

Joel Semeniuk: This is the first major Service Pack for Team Foundation Server.

Carl Franklin: Wow!

Joel Semeniuk: Yeah.

Richard Campbell: For Foundation Server?

Joel Semeniuk: Yeah. So, we've got some good stuff coming down. We've got lots and lots of performance upgrades for, like Version Control and the Work Item Tracking and the Database Warehouse kind of, as we talked about before. One of the other things that's coming out that has been a pain for some of my customers, is the ability to support different authentication mechanisms for Internet versus Extranet users. Right now, we don't -- we have to use NTLM for validation, which means that my -- people external to my organization have to VPN in, in order to get to my Team System.

Richard Campbell: Oh, right, yeah. You know, and it's interesting, it suddenly hits me like a flash. We've really not talked about what it takes to run a Team Foundation Server, because I remember in the old days, back in the early Beta, it was a real struggle to get a setup that worked; I think it was three different VPCs we were running to make that happen.

Joel Semeniuk: Yeah, it was Harry Potter magic.

Richard Campbell: Yes. And TFS definitely wants to live on its own machine; it's got its own instance of SQL Server is that the way?

Joel Semeniuk: There are many ways and they are very much governed by the conditions for which you are going to be using it. In most of my installs, we are looking at one server and we have a number of tiers, like physical tiers, that Team System has broken down into; one, the Database tier, and one is the Application tier, those are the two primary ones that we need to be concerned about. Now, you can run those guys both on the same box and most of the organizations choose to do that because with that we can support upwards of around 500 concurrent users.

Richard Campbell: That's enough for most shops.

Joel Semeniuk: Well yeah, for a lot of different shops. You'd want to start thinking about separating your Application tier from your Database tier, when you start exceeding -- there's really good recommendations and some knowledge bases around what that threshold is. But since you've started seeing, that you need to support literally maybe a thousand users or so forth, or over 500, you're going to really start thinking about separating application tier. Now there are some interesting requirements there, like for example, the application tier can't run in the 64-bit OS.

Richard Campbell: Oh, right! And I have heard noise about this, that this was a -- it's been a bit of an issue for people.

Joel Semeniuk: Right. So, if you really wanted your Database tier running with a 64-bit OS, you have to separate those tiers right off the back.

Richard Campbell: Ah!

Carl Franklin: Wow!

Joel Semeniuk: Yeah.

Richard Campbell: I see that as a mistake on the Team's part like -- it's just not that hard to make 64-bit work. I know you have to think about it; there are some things you need to do, that should have been thought about -- we are just not building non 64-bit servers anymore.

Carl Franklin: We certainly shouldn't have.

Joel Semeniuk: Well, I mean and there are other thing that you should be considering like, one of the things I would really like to see, is to take my OLAP Services and put them somewhere else. I can't do that. My OLAP Services are basically on my Data tier and I -- personally I just -- it feels dirty, I don't like that.

Richard Campbell: Well, just like you want to separate your transaction processing from your reporting. You want to keep those two things separate, so they don't impact each other.

Joel Semeniuk: Right. Exactly. So, there's also going to be tons of bug fixes in the New Service Pack. We have been using the new Office-2007, and as you know there is Excel and Project Integration with Office and it's been kind of quirky, running underneath Office-2007. So they are going to be really kind of updating it, fixing a lot of bugs to support 2007. Unfortunately no support for SharePoint- 2007; I think that's going to be a mistake as well. I really -- I'm enjoying playing around with SharePoint 2007 and I think



Joel Semeniuk Updates Us on Team System August 17, 2006

that they could really use that particular piece to extend a lot of the functionality of the tool.

Carl Franklin: Joel, what's your upcoming book called?

Joel Semeniuk: It's called '*Managing Projects in Team System*' - simple as that.

Carl Franklin: What's the publisher?

Joel Semeniuk: Microsoft Press.

Carl Franklin: Okay. When is it, what's the ETA on that?

Joel Semeniuk: I believe January, we've had to push it back more because of logistical problems than anything else but it should be released around the January timeframe and we are going to be focusing on more of the agile approach to Project Management. As you know, MSF for CMMI, is an agile-ish implementation of CMMI.

Carl Franklin: Contradiction in terms of...

Joel Semeniuk: So, we are going to be embracing a lot of those Agile principles when we are going to be dealing with the Project Management aspect.

Carl Franklin: Are you the only author?

Joel Semeniuk: No, I'm writing it with a really talented author named Martin Danner. He's had a tremendous amount of experience in the manufacturing world and when we talk about Project Management and some of the new disciplines of -- ways of thinking about production systems he is really kind of, one of the leaders in that area.

Carl Franklin: Excellent. Are you speaking in any conferences coming up here? Is there any place we can see you?

Joel Semeniuk: You know, I have been keeping it fairly low since TechEd, just doing lots of vacation and I haven't done any booking for upcoming conferences. I am doing a Code Camp though - hopefully if every thing works out right, in Edmonton, Alberta, in September. And I don't really have any major conferences booked any time soon, so.

Carl Franklin: Okay. What about other resources on the web that you might want to point people to?

Joel Semeniuk: Well, I mean obviously, I love Brian Harry's blog. He is just so transparent, in

their approach, and the product itself, I mean it's a State Book for me.

Richard Campbell: And Brian Harry is on the Team; he is a Microsoft employee building the product.

Joel Semeniuk: He is definitely on the Team. I have a guy working for me; his name is Steve Porter and he has done lot of Best Practices type stuff on his blog. So, he has a lot of Best Practices for example, around branching and merging. When do you really want to branch? When do you really want to merge? What did you stay away from? So, Steve Porter's blog is definitely one I would recommend as well. He's got a lot of really good lessons learned over the past several years.

Carl Franklin: Is he the Mac guy?

Joel Semeniuk: He is. He is the Mac guy and he is probably one of the most brilliant Team System guys I have ever met but he loves his Mac.

Carl Franklin: Nothing wrong with that?

Joel Semeniuk: But it just goes to show you that you can live in a Macworld and still have the goodness of Team Foundation Server.

Richard Campbell: That is very interesting.

Joel Semeniuk: Yeah.

Carl Franklin: All right Joel, we are coming to the end of this show and of course I'd like to ask my guest if there's anything that they've seen online that's very cool, totally unrelated or a new toy or anything cool that you've come across lately.

Joel Semeniuk: Geez! You know, I have to admit I'm not really a toy geek. I'm not sure I can think of anything that would have to be...

Carl Franklin: A game, a Podcast or anything you can think of at all, anything you want to share?

Joel Semeniuk: You know, I'm drawing a blank.

(Laughter)

Carl Franklin: Some new shoes or...

Joel Semeniuk: You know, I love my i-mate SP5.

Carl Franklin: All right, well there you go.



Joel Semeniuk: My i-mate SP5 has become my new best friend. I used to have an i-mate JASJAR and I don't like walking around with a brick in your pocket.

Richard Campbell: Because it is a brick.

Joel Semeniuk: Yeah, is that a brick in your pocket?

Richard Campbell: Except the bricks aren't so damn fragile.

Joel Semeniuk: Yeah, I actually had to send it in for repairs twice but my i-mate SP5 is this beautiful device that just works. So, I use it mostly as an output device, meaning I think with my calendar and I don't use it to enter data but it just makes me happy.

Carl Franklin: Very Good. Joel Semeniuk, thank you very much for sharing your thoughts about Team System and your insight and knowledge of course, it's very much appreciated by us and our fans. Thanks.

Joel Semeniuk: Thanks a lot guys.

Carl Franklin: All right. And we'll see you next week on .Net Rocks!

Geoff Maciolek: .NET Rocks! can be found online at www.dotnetrocks.com and at msdn.microsoft.com/dotnetrocks. NET Rocks! is edited each week by Geoff Maciolek, that's me, and Carl Franklin who is also Executive Producer. All music heard on .NET Rocks! including Toyboy, the theme song, is created and produced by Carl Franklin and Franklin Brothers Band - Carl Never Sleeps. .NET Rocks! is produced for FranklinsNet by 'Pwop Productions' - providing professional audio and podcasting services online at www.pwop.com. 'Pwop', it's time to get your impact back.

(Music)