

[ MUSIC ]

LEROY: Hello and welcome to an ebizQ.net podcast, Five Application Lifecycle Management Imperatives for Successful Global Software Development. My name is Kyle Leroy, and I'll be moderating this podcast. This podcast is being brought to you by IBM. For more information on IBM, please visit their Web site at [www.ibm.com](http://www.ibm.com).

With that, I'd like to start by introducing our expert, Andy Detandt, resource manager at lender processing services. Welcome, Andy, and thanks for joining us today. Let's jump right in to our discussion. What was your approach for implementing Rational tools to address your pain points?

DETANDT Well, in rolling out Rational tools, what else could it be? It had to be an iterative approach. You use iterative when you want to manage speed and stability with your projects, and it was never more brought to light than when I was in a Cracker Barrel restaurant some time ago.

Two gentlemen were moving through the store part of the restaurant and I heard one gentleman say to the other, that consultant is just driving me crazy, to which the other gentleman replied, I'm going to lose my lunch if he uses the word iterative again.

So iterative, iteration, sure, it's a hype word, it's a buzzword, but the only alternative is paralysis analysis when you're thinking about deploying tools across a large organization. When you put new tools in employees' hands, you naturally meet with pushback, because it's not only the tool that you're asking to change; it's also that you're asking a multitude of people to change their behavior. And it makes no sense just so simply swap out tools without bringing a productivity lift to a company.

So my advice is start small, move slowly and move deliberately. And as you get small success stories, then favorable word of mouth will spread around and be relayed around the company, and then the adoption will become more organic and natural for the various teams.

And I always advise when you are trying to do process enablement with a tool, start with a light implementation. It makes it easier for the teams to adopt. Solve only part of the problem, and then tackle the rest as you go.

Get them to come back to you for more solutions. Keep the appetite alive for innovation and automation. And if you can do so, share that credit with them because after all, they're the ones taking the risk to their project by rolling in new tools and new process.

LEROY: All right. Thanks, Andy. The next question that I kind of wanted to ask you today, How is management of work items different than just tickets?

DETANDT: Well, I think most of us are familiar with tickets, trackers, bugs, because we've lived with them longer than we have with this notion of work item. When we think about tickets, typically a ticket is just an open issue that's worked or an unspecified duration and then closed. Usually doesn't have a well-defined parameters or attributes and it lends itself to an over simplification of the issue. And the generalization of tickets is because its relation to other artifacts in the project is not explicit.

On the other hand, a work item is explicitly connected to the guts of the project or to the inner gear mechanisms of an ever-turning project that never rests. A work item is responsive to the need of the project, because it's more precise in its description of the work, and it has other supporting artifacts. And when those artifacts change, the work item adjusts to the project needs.

And in my mind, a work item has a smaller break down structure in that several work items may address a single larger issue. So thus the management of work items becomes dynamic and real-time charting of people in their work, and in this environment, you'll find the workers more

aggressively update their work items because they know the value of their contributions to the project, whereas with static tickets, it seems like it's always a chore to keep up and keep them current.

Now, if you don't change to work items, in my opinion, your projects will never be responsive to changing business needs. Business needs change in the middle of projects, and your work items related to other artifacts are able to adjust to make those changing business needs.

Work items in my opinion are that fundamental mechanism around which people self organize in order to speed work and improve quality. And I've found that in this environment, it seems like the team members are much happier because a unit of work is communicated to that person, they acknowledge it, they understand it and then they complete it, upon which they contribute it back to the team and then that team evaluates it. So this then is also a good, sound basis for smart resource management.

Without work items, you really can't have an ALM solution. If you think about it, if you have ever heard executives say, now, tell me again why it takes so many people to do this work? With active management of work items, that executive will finally have a true, accurate answer at any given moment.

And tickets just don't cut it because comprehending the total significance of a list of dissimilar tickets, it's just guesswork. And that's my answer and I'm sticking to it.

LEROY:           Andy, thank you for your answers. This has been an interesting and informative discussion. Thank you for your time today. And thanks to our listeners for taking time out of their day. I would like to thank IBM for bringing us this ebizQ.net podcast, and thank you all so much for joining us.

[ MUSIC ] [END OF SEGMENT]