The Napkin Guys: Reasons for ERP Change

Bryan Bechtoldt: Hey buddy, how are you?

Tim Hanson: Good, how are you?

Bryan: Good.

Tim: Cooking up some dinner for us

Bryan: Yeah

Tim: Thanks for having us over.

Bryan: Yeah, the dogs for the kids are just about done. Our stuff's looking pretty good.

Tim: It's lookin' good.

Bryan: It's not too bad.

Tim: Good.

Bryan: So how you doing?

Tim: Good, how are you?

Bryan: I'm okay, I'm okay.

Tim: Good.

Bryan: Hey, listen man. I got a buddy, he kind of cornered me the other day. He's got a growing business that's just going gangbusters and he's got kind of a legacy ERP kind of accounting software system and you know it sort of kind of does what it does but he really doesn't feel like it's giving him you know kind of the ability to compete with some of those competitors they've got some newer systems and he thinks they have an advantage and I don't know if he's right or he's wrong but I'd like to like give him some good advice.

Tim: Classic, classic problem. You know, compelling reasons for change in ERP especially when you compare these to the traditional ERP solutions.

Bryan: Yeah.

Tim: You know, let me give you a couple of topics and see if these work for you. You know one of the things that's challenging is, most organizations today have the core in place.

Bryan: Right.

Tim: So, you know, dealing with accounts payable, and accounts receivable, and general ledger.

Bryan: Really the accounting stuff, right?

Tim: And even the inventory, and order processing, and manufacturing software in general hasn't really changed. So, going on the market, those feature sets on the market today are comparable to feature sets that have been on the market for years.

Bryan: Okay.

Tim: More what we look for today is, you know, how does an organization build, kind of, this end to end solution given the fact that they may have a mobile workforce, maybe have, you know, different point systems that come to play. So, a couple of things that maybe I'd point you down. One, look for something in the product today that's device independent. By that I mean…

Bryan: Oh, you're giving me a list. Can I have your napkin?

Tim: Yeah.

Bryan: Thanks, buddy.

Tim: So device independence. So the reason for that, Bryan, is that many organizations they have a mobile workforce. They might have a workforce that's in different offices.

Bryan: Yeah.

Tim: They have a workforce with people working on laptops, iPads, smartphones.

Bryan: Sure.

Tim: They need an application that can scale to whatever device and be used by that device wherever they're located.

Bryan: So people can kind of pick up the information where they're at, they're not really attached to the office that much anymore.

Tim: Exactly.

Bryan: Okay.

Tim: Second thing that I would look for is, because most organizations have built systems that have kind of different systems in place that might have manual steps that need to be performed. They're looking for ways to be much more efficient as an

Organization. So they're often looking for ways to put automation in areas that weren't always there.

Bryan: Ahh, I was looking for the point there, automation.

Tim: And I would look for solutions today that have an automation engine.

Bryan: Yeah.

Tim: Often that automation engine is rules-based. We can build rules that if this happens, automate this task, perform this function. Sometimes those functions don't have to be core functions of the software, they can be functions outside of the software.

Bryan: So it's almost kind of like a workflow.

Tim: Exactly, like a workflow. A third key thing that I'd look for is, you know, an organization that keeps - I'm sorry, an application that keeps the organization informed.

Bryan: Mm-hmm.

Tim: We would use the term like a notification engine, an alerts system, an approval engine. Simple things that people are familiar with might be, if I'm a product distributor and I ship a product, I want to notify my customer that product shipped and I often want to notify them via email, but it may be more complicated than that. It may be something like, you know, if a customer places an order in this geographic region and that order is over a certain dollar value, alert the territory manager.

Bryan: So messaging the key events to key personnel.

Tim: Exactly, keeping everybody informed.

Bryan: Yeah driving the business, right?

Tim: Yeah. And then the last thing, I think, would be more of a technical feature. Something, I'd use the term SOA.

Bryan: Service-oriented architecture

Tim: A service-oriented architecture.

Bryan: That's like a bread basket for me.

Tim: Well and you and I could talk. We can talk probably a different time on that one because that one we could talk for hours on, but one of the advantages of a service-oriented architecture is it lets us take different pieces of software that might be really strategic and point to the organization,

Bryan: Yep.

Tim: and use that bus strategy, SOA strategy, for connecting those applications. Again, more technical but I think a key feature of an end to end solution that really starts to differentiate, you know, why would somebody change.

Bryan: Yeah, it's really one complete application, one complete system.

Tim: Yeah.

Bryan: And really there's no application silos that exist.

Tim: Yeah, those are four points. Let me see the list.

Bryan: Yeah, there you go. My chicken scratch is bad, man.

Tim: Yeah, I think this is a good list. I mean, if your client looked for applications,

Bryan: Yep

Tim: That are device independent,

Bryan: Yep

Tim: Allowing that mobile workforce to really be connected,

Bryan: Yep

Tim: An automation engine that could make the organization much more efficient, build rules-based automation logic, notification alerts approval system

Bryan: Right.

Tim: And then, one fundamentally built on a service-oriented architecture.

Bryan: Yeah, sounds good. Well that sounds great.

Tim: How's lunch coming?

Bryan: Can I have that back?

Tim: Yeah, you can have that.