

John Rhodes ADC Austin Interview



Mark Schroeder:

I've created a list of questions that are important for companies when they're thinking about modernizing their current applications. Today we're going to talk with John Rhodes from ADC Austin who is an expert in the area of modernization. Then we'll also look and see what kind of solution ADC Austin provides for modernization. John, give me a short bit of information about your background in software design and modernization.

John Rhodes:

Sure Mark... my background started several years ago. Just when I was fresh out of school, I joined with Kraft foods and my first project there was to help modernize processing logistics from the main frame to the IBM midrange platform. I kind of dived into it right away and was able to see this process was bringing a lot of benefits to the business by producing lower-cost, operational costs and also increasing productivity and capability of the software. So in addition to the benefits I could also see the challenges. So it kind of got me hooked. When I later joined CNA it was also a key focus of my job to help people modernize old applications with the new technology. So I decided to make that a professional focus area. One area that I've been interested in for several years is automated migration of Legacy software as having lot of promise, but also being very challenging. In recent

years we're seeing a lot of focus on that and seeing a lot of new technology and tooling coming into forefront and we actually focus on that at ADCAustin. So it's an area that has a lot of promise and a lot of challenges, but you can tell by the number of COBAL - RPG based shops that it is a major focus. For example, IBM 70% of the world's financial transactions run through COBAL based systems. This gives you an idea of the size of the opportunity.

Mark Schroeder:

That is quite a bit of opportunity with that many transactions going through COBAL. Most of these applications, people would consider to be Legacy applications. What is your definition?

John Rhodes:

Yes, well that's a really good question. Obviously, Legacy can mean a lot of different things to different people. Some people consider that to be any production enabled system. Probably a more common definition is a system that is out dated from a technology perspective but still contains considerable business value.... Or to put it a different way, a system that the software developers and businesses would like to replace it if they could, but they can't figure out how to do it cost effectively.

Mark Schroeder:

That's a problem with companies. It's such a costly endeavour to change their entire application to something that's more modern. Either replace it or find some other way to manage their software. A lot of companies are looking and saying they want to modernize their software... How do you define software modernization? Is it changing the code? Is it changing the platform you're on or everything else?

John Rhodes:

It could be all those things. Broadly speaking it's the process of evolving legacy code into more modern software environments where these applications can be maintained more effectively. In days past that would mean rewriting the code. Taking the original COBOL system and writing that in Java manually. However, with current technology we're seeing more automation and again, that's our focus at ADCAustin. It's how do we apply automated tooling to this problem?

Mark Schroeder:

Automated tooling seems like it would speed up the process quite a bit. Maybe even make better applications.... because I've seen when you use a tool, you have to do all this coding manually. There's a lot of manual mistakes that get put into the process.

John Rhodes:

Absolutely when you're talking about a 20 million line system it's impossible to rewrite that manually without making some mistakes and just a small number of mistakes can mean a large effect on the business. If those mistakes are in the area of calculating a stock price or the value of the transaction for example.

Mark Schroeder:

One of the reasons I think companies modernize is because they have all these different applications out there and they want to find a way to manage it. Why is it important to consider the whole portfolio of applications that a company has when looking at a modernization approach?

John Rhodes:

You're right... Enterprise applications don't operate as Islands or standalone systems. They are interacting with other systems and sometimes interaction can be quite deep so you really have to look at the entire portfolio of software assets before you decide to modernize and go forward. It does no good to modernize the accounting system if ERP system just can't talk to it.

Mark Schroeder:

I think we can learn from our government. They have many different branches and their systems don't all talk to each other... do they?

John Rhodes:

Sure, the government is an organization that is very complex and difficult to modernize.

Mark Schroeder:

When companies are considering modernization what are some of the reasons you see companies need to modernize their applications?

John Rhodes:

I'd say the top reason that we encounter is basically that the legacy software system cannot be kept up to date or maintained at a pace that is acceptable to business. Perhaps a close second is that the legacy software cannot be integrated effectively with other enterprise software. Those two reasons are really the top reasons. Some other reasons that come into play could be a need

to or desire to move to different platforms or explore different platforms.

Mark Schroeder:

You mentioned integrating applications. Why would an integration actually be a motivator for modernization?

John Rhodes:

Well it's basically a fact that most or many legacy applications can't be effectively integrated with other modern software. They don't have an SOA.. , web service bus for example. They don't have the mechanisms to get in and automate parts of the code. They operate in kind of a black box that is opaque to the rest of the enterprise. So when a company purchases for example an enterprise business process plus modernization solution and then they find that they can't get the benefit of the solution because they just can't get into the legacy software. That can be a powerful motivator to modernize.... to modernize legacy software to get the benefit of the other software that you have in the organization.

Mark Schroeder:

Then I think one of the other things you see is that all these other software sit on the other platforms and your legacy application

may sit on something that doesn't talk very well with it. What about considering moving your application to another platform as a motivator for modernization. Have you seen this as something that companies are trying to do and consolidate in a certain platform or maybe expand to certain platforms that talk better together?

John Rhodes:

Yeah absolutely it is a key driver especially for the software vendor so a company that business is developing software... You know they have to as a matter of survival provide new and expanded offerings to their customers and??(7min39sec). The fact is the market for 5250 green screen applications is shrinking. On the other hand the .Net, Java and Lenox platforms are becoming increasingly more important. So we have several high speeds that are working with us in looking at migration from their RPG or COBOL systems and seeing that it is just a way to stay in business. Then there's also demand outside the IC supporting multiple platforms can be critical. Companies may find themselves through acquisitions or other reasons that they have a wide variety of platforms to support from mainframe to midrange to UNIX and standardizing those platforms into one enterprise platform that is around a single operating system and a single technology can provide some benefits. So those are the two main reasons we see that organizations are looking at re-platforming.

Mark Schroeder:

Do you find that they are mostly moving to a single platform or do they move to multiple platforms?

John Rhodes:

Well for the ISV, supporting multiple platforms could be critical. Obviously if you have a great application around flow manufacturing, you want to have that available to people who want to run it on Windows.net as well as people that are interested in Java and not have the platform as a barrier to your success. So that can be a reason to expand other platforms. I think I see more consolidation of platforms than the desire to introduce new platforms for your typical business organization.

Mark Schroeder:

Of course we always have the government to deal with don't we? In our applications especially the larger applications, do you see that regulatory compliance have effects on companies that want to modernize and why would compliance issues make companies want to modernize?

John Rhodes:

Well you certainly described a very powerful motivation. You have to as a business stay with some legal operating requirements. You know the government can quite easily shut you down. Government regulation is very important and if an organization finds they just can't modify their systems quickly enough to meet new regulations or it's costing too much they don't have enough quality around that. It's a serious risk to the business. That's especially true now. We are in a period change in regulations. We're seeing new financial control regulations come down from the federal government. Changes in healthcare, cap and trade bill. All these things are resulting in more regulation, not less to change systems faster. Typically we see some of our customers at the state and local level having to put systems and procedures in place to handle the fiscal stimulus bill. They're being told, "here's some money for you to spend but in return you have to report use of this money in a much more stringent way". So that's driving some modernization activity as well.

Mark Schroeder:

That seems like that could really provide some issues for some companies that don't have many programmers or that use 3GL to rebuild a lot of their application. To me the 4GL is really going to help those smaller companies to modernize more quickly and meet the demands.

John Rhodes:

I absolutely agree.

Mark Schroeder:

Another thing that companies are trying to do... it's such a competitive world out there you have to find ways to have a competitive advantage over someone. So they're modernizing their processes... maybe streamlining what they need to do to be effective. How does this whole area of process improve when it's a modernization effect? The modernization strategy of the company?

John Rhodes:

Well yes it is very important of course. Business agility the ability to take in new business concept and integrate that into business processes and do it quickly is very important to a lot of companies. When you're talking legacy software you often have a situation where specific transactions and other information can be buried within a system through hierarchal menus and you can't rearrange your COBOL, RPG based screens in different sequences or you can't access something else in the application. What happens is these Legacy applications are not work flow enabled... so business process modelling, concepts and business process of change just can't be done effectively or quickly

enough, so that's another motivator for modernization's to basically support business that it should be supporting.

Mark Schroeder:

You've worked with some companies that provide some workflow capabilities built right into the applications. Is that correct?

John Rhodes:

That is correct. We're a business partner with SORCO. They provide so-called expert services that helps organizations basically bring together computer systems through a common workflow environment. We work with other tools too like Lombardi Teamworks and we feel quite strongly that this type of software is important for businesses to keep up with demand of changing business processes.

Mark Schroeder:

It seems that is as important as modernizing your application because you're helping the application be able to follow what the business is really doing.

John Rhodes:

Absolutely

Mark Schroeder:

Another thing that I've seen over time as I read the journals about modernization out there is that developer's skills are having some effect on what goes on with modernization. Especially you know I come from the I series world and it's a little different there versus the DOS, Well, it is a critical area...you've heard Java or .net world. How do the skills of companies affect their modernization strategy?

John Rhodes:

Yes, it is a critical area. The average age of the RPG builder is 55 years. We all know that universities are not offering courses in COBOL, RPG on their standard curriculum. So you don't have to be a genius to understand that these skills are getting harder to find. On the other hand Java and .net skills are getting easier to find. That is another reason that people really have to look at their outdated portfolios... As these programmers start retiring it's going to be harder.

Mark Schroeder:

It's amazing that COBOL after all these years is such a strong language out there. There's no place to learn that it seems like unless you take a self-study course on the Internet.

John Rhodes:

That's correct. Some of our customers are adding COBOL at many thousand lines per year. The problem keeps growing.

Mark Schroeder:

Companies have different approaches to modernization out there and each **one** has its reasons to do it and not to do it. What are the different approaches that you've come across?

John Rhodes:

Yeah, I suppose the classic way to modernize has always been to rewrite the code manually. So to take the Legacy system and sit down and get the business rules out of that and redesign an industry right? We're finding many of our customers simply just can't find value in this anymore. These large software projects can span years. It just takes too long and by the time the system is finished the business has moved on. So that's becoming less and less of an option for people. Purchasing package software is another way to move to more modern environments and that can work well in a lot of cases but in the case of many businesses that

the software may not fit the operative environments. In that case the cost of modernization can outweigh the benefits. You can wrap the legacy code in a way such that you can get to SOA or perhaps using screen scraping techniques to put a front end. Those are all quite common. In the case of screen scraping you can get a solution. A modern front end to your legacy application that runs on the web. However since you're not modernizing underlying business logic the effect is almost skin deep in some cases and you have the same problems that you had before around business agility and changing the software to meet business needs. The last techniques that were seeing rising in popularity is SINTAX translation. Which is taking the for example COBOL code and translating it into Java. One reality with this however is that you often end up with an unmaintainable mess at the end. So that your COBOL system is indeed transferred into Java but you wind up with the JOBAL system as some people put it back either the COBOL programmers or Java programmers can effectively maintain. What we're focused on is various syntax translations where we extract the design information at the modern level and basically translate that into a second modernized level. So with our approach because we're not translating individual lines of code were able to give you something that's maintainable at the end after it's modernize.

Mark Schroeder:

That's an important factor because maintenance of a system really is the largest, time-consuming function in most cases because that system has to last for years.

John Rhodes:

I have to agree... Certainly if you have a case where a software system is very static and doesn't need to be maintained ...then perhaps a pure syntax translation is appropriate. In the case where you are using something or intend to maintain it for five or 10 years in the future than a pure syntax translation won't work.

Mark Schroeder:

you start talking a little bit about the approach about your syntax translation I'm assuming that's what you include in your M3 approach. Could you explain a little more in depth about the ADC Austin M3 approach you developed?

John Rhodes:

Absolutely... Basically our M3 approach stands for model-based modernization. So we're focused around several models that generate code. Specifically we're focused on the CA model-based development customer. So that means a customer that using a tool called CA2E or sign on ... with our modernization we take that software model and translate that into a different model of CA

Plex and from there we generate C. sharp or Java. Additionally we have a tool called data growing the analysis as a modern base system for RPG and COBOL what into the analysis does is take the RPG or COBOL and extract the business rules into an expository model and then use that to generate code from. Both these tools we feel provide a more complete and maintainable modernization method because again or not modernizing a single line of code line by line. We're modernizing the model and business rules.

Mark Schroeder:

That sounds like quite a unique approach. Using models to actually manage and generate your code for the future. Are there other companies besides ADC that use this approach or is this something unique to you?

John Rhodes:

I wouldn't say it's unique to us, but it is not made the most common way certainly. It's something that other companies do offer with different tooling. I say that our advantage at ADC comes from a deep understanding and specific tooling around the CA and the IBM customer. So we have some specialized software around that. So if you're in that environment we feel that we offer the highest degree of automation and we also deliver more quality and maintain one product.

Mark Schroeder:

If I understand.. You can take an application from your green screen all the way to become a web enabled Ajax application. You use....I think Web city or a web client to do this. Could you explain how this works and when you're doing this whole conversion from the 400 green screen to the web do you have to rewrite a lot of code?

John Rhodes

That's a good question. It certainly does vary according to specific circumstances how the original code of was written and architected. What we've tried to do and we believe we've succeeded in is to take the original 2E model and apply a very high degree of automation to it. So we can take a running function run through automated processes and implement it in and generate a web application directly from without having to make any modifications. We don't have 100% solution of course and that's our intention and our approach is to have a high degree of automation that will approach 100%

Mark Schroeder:

Well that can really save a company a lot of time and money if they can have 95% to 100% of their code moved over from the green screen over to the web.

John Rhodes:

Well that's absolutely true. Some of the systems that we look at can range from 10,000 to 15,000 functions that may be anywhere between 3000 and 5000 individual panels and if you can't automatically port those and say you have to spend a minimal amount of eight hours per screen or function. That adds up to some big numbers pretty fast. We're not saying we have the perfect solution but we are saying that we have something that allows you to cut down the amount of time you have to spend after this is modernized to a minimal.

Mark Schroeder:

In fact you could probably be as efficient or more efficient in your conversion as a company could be in providing screen scraping.

John Rhodes:

Yes, that's the intention and the reason we've invested so heavily in this technology.

Mark Schroeder:

One of the buzzwords out there nowadays is Ajax. Everybody wants to have an Ajax enabled application and have all that gooey stuff going on the web. Does the M3 approach include Ajax capabilities?

John Rhodes:

Yes, it absolutely does. We can see that Ajax is taking off. I've recently saw a statistic that most organizations have RA applications in production. We do offer that as part of our platform the modernize application can be generated as an Ajax application with all the nice features you find an Ajax, like a smooth scrolling on grids, interactive charting and things like that.

Mark Schroeder:

That's very nice to have that Ajax capability right there. Do you have to actually write the Ajax code manually or does your process generate the Ajax map version process?

John Rhodes:

Yes, we have two actually generate the Ajax from the model. Ajax coding can be very complicated and also if you don't do it the right way you can induce security holes or have other problems so we

are able to provide a general solution in that a few eyeballed the full-featured output.

Mark Schroeder:

One of the other issues that I've found in software development is that as you're developing you have multiple developers working on a project and someone might be working on one function and another person on another function. Then the whole change process becomes an issue. The M3 approach, does it include some capabilities for change management or is it something that has to be handled manually?

John Rhodes:

Good question... Application management is critical to the modern software development process. We certainly have that as part of our technology so when we do modernization we format using enterprise tooling such as MPS integrity and then we implement that with the customer as a by-product of the modernization process.

Mark Schroeder:

Your M3 approach. You've talked about using it in a model-based system. Are you trying to target specific organizations or types of companies that would use the modernization approach?

John Rhodes:

Our customers span virtually all industries. We have customers and manufacturing, finance, higher education, government. In that respect we support every one. Probably the most significant segment for us is other software vendors that really have a compelling need to modernize and expand their markets through modernization .So we have some specific support for the enterprise ISP. We do support multiple platforms during the modernization so that the ISP can generate AA, Java, or .Net. And also we support multilingual applications so that enterprise applications may be sold worldwide. Maybe to include for example Japanese, German, English, and Spanish versions. So we have some specific things around that that we can target and we do also focus he is primarily on the CA model-based customer. Those are people that are using CA, 2E and are not untypical.

Mark Schroeder:

Well since you mentioned Scion 2E I've worked with it myself and I know you've worked with it quite a bit. A lot of people may not know exactly what 2E does. Can you give me a brief overview of what CA2E is?

John Rhodes:

Absolutely, CA2E was originally known as Scion 2E. It was one of the most successful and still is one of the most successful coding generation platforms out there. It was developed for use in the system 38 when the system 38 first came out. Now it's called the IBM I. And in it enabled developers and companies to develop software far faster than they are able to do in using hand coded RPG and COBOL and it is a model-based tool so when you're developing your applications in a model and generating code. However, it was quite successful in this day and is showing some signs of age now. You are limited to generating in RPG and COBOL. You're limited to presenting your presentation layer in a green screen environment. So almost by definition it can now be considered legacy. People are looking at what they need to do to modernize these systems.

Mark Schroeder:

One of the partners to 2E is CA plaques. That's an application that's been around for a while now and also what's the difference CA plaques and CA 2E?

John Rhodes:

Well, CA Plex was developed by CA to be the modern equivalent or replacement for 2E. So CA Plex is a graphical model-based development environment that enables you to generate from a high level business model modern languages like C# and Java.

CA Plex also has concepts like design patterns and inheritance. Things you don't find directly in CA2 E. So it's very similar to 2E but it's also a step above. It's a modern equivalent.

Mark Schroeder:

Sounds like it's an interesting environment to develop in. Does that provide benefits over a 3GL?

John Rhodes:

Yet we think it does provide some significant benefits. This CA Plex has the ability to use higher-level software patterns and design extraction and using that capability provides some very complex software models and then you don't have to implement the models by hand. You actually generate the models from the tool yourself. You either generate Java or .net or both. So you're able to get a degree of platform independence. With Plex one developer can do the work of several hand developers.

Mark Schroeder:

That in itself just by having one person do the work of multiple hand coders probably saves a company quite a bit of money.

John Rhodes:

Yes, that's absolutely the case.

Mark Schroeder:

I think that's one of the things we always have to look at is what kind of cost are we putting into something and what kind of benefit we are getting out of it.

John Rhodes:

I agree.

Mark Schroeder:

Both CAPlex and CA2E you mentioned are model-based. Is this a new way of developing applications or has it been around for a while?

John Rhodes:

Model-based development is not new. It has been around for a while. At least 20 years but it's gaining some popularity recently as a better way to develop software in many cases. We're starting to see some new tooling from IBM and Microsoft are in where they're producing model-based development platforms or their modelling tools are developing some model-based capabilities.

Their trivia on rails as a model-based capability has gained some popularity.

Mark Schroeder

You also mentioned that with Plex there is what are called software patterns. How does Plex use the software patterns and what benefit are they for developing software?

John Rhodes:

A software design pattern can be considered to be a blueprint or template for how to solve a software problem in different situations. One of the more common examples is a master D-cell pattern. You have mastered D-cell patterns and applied invoices where you have an invoice header, invoice lines and maybe a contact list. The contact list owner and details and really an endless variety of this pattern but they all have the same needs to have a single master with many details and the capability to maintain and operate on these as a unit. What CA Plex gives you is the capability to take this abstract concept and basically encode this into a model and then use that as CA Plex code generation capability to apply that model to this specific classes and specific situations in a software system. This can be a very highly productive way to develop new software because you're not constantly creating clones of the same concept.

Mark Schroeder:

You've been talking about CA Plex and CA 2E working with model-based development and how they increase the productivity that a single programmer can do. It seems to me that a company would want their programmers to be more productive so why hasn't model-based development been adopted more often than it has?

John Rhodes:

I think there are a few reasons behind that. I think the main reason is it does take time from investment and abstract design and pattern-based methodologies to really get the most out of model-based development systems. It's not something you just dive into and suddenly start being very productive with but we found that our customers that are willing to make the investment and are willing to put people on a project that have abstract thinking. The payoff can be pretty substantial because again you're creating a set of abstract patterns that you've applied across a situation so one person who has the proper training and proper mind-set, can be as productive as 3 to 5 hand developers.

Mark Schroeder:

You have some large organizations that have some 2E applications that they have been developing over years and years. Maybe 10 to 20 years of development. Why would these

companies consider moving to CAPlex to modernize their applications versus maybe EGL or some of the other options out there?

John Rhodes:

Well.... I guess primarily the reason is that CAPlex in many ways is similar to CA2E. So there are a lot of benefits behind that. One benefit is that the learning curve is relatively gentle. The two 2E developer that has training in the Plex tool can come up to speed pretty quickly because the software looks very similar to what was in 2E. So they find things to work on and be productive immediately. The other thing is that the investment that people have made in their 2E model's is considerable and with our approach are able to preserve that investment to a high degree so the business logic and the model structures that you have designed in 2E come across a high degree of automation and CA Plex and look the same and operate the same. This makes it a very productive environment and preserves the high degree of investment we've made to your model. The last reason is that CA Plex can generate not only C. sharp and Java code that you would expect from a modern tool but it can also generate RPG. That includes DBS green screens and reports. This can be important in that you know I sure moving forward to new modern development techniques. ^ Java and .net you also need to support your older code that is in RPG or COBOL. So Plex can

generate RPG, can generate green screens to help you transition your way into the modern development easier.

Mark Schroeder:

Another one of the buzzwords out there that you hear and you read about all the time in the software journals is SOA and how many organizations are moving to use SOA. Does ADCA use SOA and can they support it with Plex?

John Rhodes:

Absolutely, our customers certainly demand that as customers do everywhere. Typically we do you provide an SOA service bus as a by- product of modernization so that you're able to access your modernize business logic through Web services for example or embed your new screens in business process model layers. Absolutely, we find this a very important and compelling reason to modernize.

Mark Schroeder:

Plex.... does it actually generate Web services for Java or .Net, or would you have to go down to using some of the more hand written tools to integrate Web services into your Plex environment?

John Rhodes:

Yes, because Plex is multiplatform, it does do well at supporting both Java and .Net. Plex has a native ability to generate WCF Web services so that you can create these .Net style Web service access to your existing Plex functionality and that's out-of-the-box. There's also a traditional tooling by third-party vendors that people such as ourselves have provided. A Java-based with a service solution. Of course since we're talking about Web services there is a degree of and it doesn't matter if your Web services.net or Java. The end result the interface can be accessed by a multitude of different operating systems and applications.

Mark Schroeder:

You mentioned that you work together with CA, Data Borough and Web City was mentioned. Why did you end up working with these companies instead of developing the whole approach yourself?

John Rhodes:

Well, at ADC Austin we see ourselves more as a modernization solution provider and not really as a tool vendor. We offer screen software tooling certainly, but we don't have to have that, as

internal we developed. We are willing to go outside the company and find and the best-of-breed software that meets the needs of our customers. When were looking at a modernization project there often many different aspects to it. We may provide a piece of it like the actual 2E , CA Plex modernization or where customers for example have COBOL code or RPG Code that is outside the 2E model. We will certainly employ solutions from different vendors to get the best solution for our customer.

Mark Schroeder:

One of the big issues I think that companies... especially smaller companies.... have when they're going through the modernization process is the need for support because they don't have all the skills on hand. So what kind of support do you at ADC provide during, after and actually before the process even starts?

John Rhodes:

Well, that's a good question. It is certainly with modernization. It's not just porting the actual code from one environment to another. In fact that's often a relatively small part of the project. Our service typically starts with the discovery process where we help the customer analyse their software with automated tooling and through perhaps in Databorough Xanalysis as an actually discover and find out what it is they need to migrate and what it is that they don't need to migrate. As we move into the

modernization step in addition to actually migrating the code we also provide training and project management. We try to provide a complete service package to help our customers move from one platform into another in a holistic way

Mark Schroeder:

Sounds like you pretty much cover every aspect of helping a company modernize their software. A lot of times you'll go to a vendor there and they'll do one piece or another piece and you have to go to multiple vendors then to figure out what to do. Now if someone wanted to find out more information about your products or if they just want to learn about what you have, what is the best way to contact you or contact ADC?

John Rhodes:

We certainly have a great deal of information on our websites. We have two websites ADCAustin.com, [ADCAustin tech.com](http://ADCAustin.tech.com). The ADCAustin.com website is very informative and go out and research about CA 2E and CA Plex. There's also a couple more independent independently maintained. We also have just launched a new site called M3modernization.com and that explains our modernization methodology in more detail and has some target information specifically placed around modernization.

Mark Schroeder:

Thank you John, for your time. It's been very educational.

John Rhodes:

That's fine Mark... I thank you for the opportunity to talk with your audience.