

ARCHITECTURAL DRIVEN MODERNIZATION

*Mark Schroeder interviews William
Ulrich and Philip Newcomb*



William Ulrich

William Ulrich is President and Founder of the [Tactical Strategy Group, Inc. \(TSG\)](#). TSG is a management-consulting firm that specializes in information and business planning strategies. Mr. Ulrich, who is also a senior consultant with the [Cutter Consortium](#), has more than 30 years of experience in the information management field. His system transformation strategies have been the basis for application migration projects worldwide. Mr. Ulrich currently serves as Co-chair of the [Object Management Group \(OMG\) Architecture-Driven Modernization Task Force](#), Co-Chair of the Business Architecture Conference Series, Co-Chair of the OMG Business Architecture Working Group, Editorial Director of the Business Architecture Institute and is on the Board of the Business Architecture Society.

Philip Newcomb

Philip Newcomb is CEO of [The Software Revolution, Inc.](#) and an internationally recognized expert in the application of artificial intelligence and formal methods to software engineering. Over the course of 32 years, he has done groundbreaking research in the application of artificial intelligence, software engineering, automatic programming and formal methods technology to industry software problems.

Mark Schroeder interviews William Ulrich and Philip Newcomb

Mark Schroeder:

Today we're going to be talking with William Ulrich and Philip Newcomb about software modernization. First of all, I'd like to find out, what are some of the software issues that are common to many business organizations today?

William Ulrich:

Well, I'll start out with that one. One of the challenges that organizations find is that the systems grew up in an age that both organizationally and technologically date them. So if the systems are 15, 20, 25, 30, 35, 40, years old, they're going to reflect the environment that they grew up in. So.. If it's from a software technology perspective that's probably going to be older languages, potentially older platforms, older databases and so on. From a business perspective they'll probably be much more reflective of the older organizational structures. So, if those structures change they may be out of sync with the way that the organization works or should be working today.

Mark Schroeder:

One of the options that a lot of companies look at when they start approaching modernization, is they look at doing a complete rewrite. Why is that a good approach, or is that a good approach, for companies to take?

Philip Newcomb:

I'll take that one on. I think that in general a complete rewrite is missing an opportunity to take a look at what's good about the existing Legacy systems. After all, over many years organizations build up those systems and the maintenance that they put into them constitutes a growth in the value of that information system as an asset. A complete rewrite is like, almost like starting over from scratch. Unless you're information business processes change so radically that there is absolutely nothing of value in the Legacy system, and the rewrite approach is going to be a lot more expensive than the alternative, which could be modernization of the information system.

William Ulrich:

I'll just follow that up with one quick comment. In Chapter 1 of our book, we got some statistics on the cost of rewrites and if you're really, really good you might be able to get it down to around \$18-\$20 per line of code. If you're more average... A 1 million line system it's going to cost you 15, 20 or 20, \$25 million dollars to

write. So, if you're talking about 5 million lines, you can start extrapolating those numbers out. We also have some statistics about the length of time it takes to develop these systems, the skills that it takes to rebuild all the business rules as Philip talked about, the business processes and support those. What you end up with very often times is the situation that where the system doesn't satisfy the business requirements even if you do deliver it. More often than not, it's not delivered on time and on budget. We have some statistics on that as well.

Philip Newcomb:

Yes, I'll just give you an example from real life. There's a system called Vista, that the Veteran Health Administration uses through all of its health, hospital systems. It's about 2.8 million lines of MUMPS. In 2008, the V.A. estimated that it would cost \$11 billion dollars to manually rewrite that system, just to give you an idea of what the manual rewrite costs would be.

Mark Schroeder:

That's quite a big chunk of change there isn't it?

Philip Newcomb:

Yeah..

Mark Schroeder:

Aside from the change though, it's also a long time that they take to build something like that.

Philip Newcomb:

Well yeah, and also they built the system by what they call the Confederated approach, which meant that the hospitals, the 168 VA hospitals, participated in development of that system. So it turns out that from a healthcare perspective it's a good system. From a technical perspective of being in the MUMPS language it is a tremendous liability for carrying the system forward to support its integration with other systems like MHS, Military Health Care systems and with other systems of the veteran healthcare. So as new standards arise and it has to be adapted, there's just not all that many MUMPS programmers around, and the cost is extremely, extremely high. The cost of maintaining that system since 2000 has been \$2.5 billion. That's another huge problem that organizations have. The enormous cost of maintenance, very small number of vendors who can provide for instance, MUMPS maintenance services and an enormous cost for manual replacement.

Mark Schroeder:

That brings up the next question. When companies are looking at modernization, I'm sure there are some certain questions that they need to look at and answer in order to be able to move forward and determine what direction they should take. What are some of these questions?

William Ulrich:

They should definitely look at the landscape and determine if there are tools and service providers available to help them carry out the modernization of their systems. Trying to do it yourself is one of the surest paths to failure. This is because information systems are enormously complex, we're dealing with the language themselves and there have been so many failures in the past, by even tool companies developing modernization technologies that were not all that good. They ought to look around and find the ones that can obviously do the job well and begin working with them. I think they are going to find that's a lot more effective for them than if they try to do it on their own. That's one of the risks.

Philip Newcomb:

... And just to follow up on that. One of the things that shocks me all the time is the lack of knowledge that organizations have about what kind of options you have with existing systems. You actually can do numerous things with the existing systems in terms of

modernization. A lot of times there is a lack of education, so it's almost a pre-question of have you educated yourself and your management about what some of your options are? Then they'll quickly figure out, as they learn what's out there, which again not that many people are that knowledgeable about it, as they learn what's out there they'll realize it's very unlikely that they have a lot of the in-house skills and certainly they won't have the kind of tools you want in the house that you need to have to do this work, so there's a big educational issue on this matter.

Philip Newcomb:

Another thing too, I would say that I've seen is a problem, is that they do need to look around and find you know, what are the good sources of information about how they can carry out modernization, because things have changed from 10 or 15 years ago. Some of the advice that they might be getting from historic forces is still I think sometimes in the dark ages. Things have changed and there are now highly effective automated approaches for information system modernization and organizations should be taking advantage of all the advances in technology that have come along. You know it used to be that test programs were pretty poor but today test programs just routinely beat masters. In fact they beat the Grandmasters. In fact that is the kind of evolution that has occurred in automated modernization technologies.

Mark Schroeder:

Now you guys specialize in a certain approach to modernization that's called Architecture Driven Modernization. Can you explain to me, what is that?

Philip Newcomb:

Bill, why don't you take that. You're the co-chair of the ADMTF or the OMG.

William Ulrich:

O.K., thanks Philip...The concept of Architecture Driven Modernization or ADM, was introduced a few years ago because there is a concept where you can just use modernization in a real tactical maintenance programmer perspective and gained some marginal benefits on it. The reality of it is that modernization is an extremely powerful concept if you can look at things...When I say things I mean the business, the systems, the applications, the data, the underlying technology architectures. If you can look at all those from an architectural perspective, determine the architectures that you really want to achieve from all of those perspectives and then use modernization to move from where you are to that future state. So we introduced the term Architecture Driven Modernization to really let people know that there's a much

more strategic view of this that goes way beyond as Philip was saying, some of the thought that permeated the industry 10 or 15 years ago where you'd fix some code maybe to improve a program here or there to improve maintenance.

Philip Newcomb:

I'll just add to that, that another of the significant advancements that has occurred through Architecture Driven Modernization, especially it's becoming a task force within the OMG. It's a very adaptable approach to an organizations need. In the past, I think especially back in the 1980s and 90s, we tended to have cookie cutter kind of tools and ADM by definition adapts to the needs of the organization. The notion that it's architecture driven is that as part of the modernization process, we take into complete account the customer's architectural objectives and adapting the technical solutions to their specific needs.

Mark Schroeder:

I think a lot of companies, what they do is they look at the tools and purchase a tool and then have to make that fit into their modernization approach.. So this is going from the other way. You're doing your planning before hand if I understand correctly?

William Ulrich:

Well, we found that it really is the only successful way... is to understand what you're trying to do and then a solution will emerge for what you are trying to accomplish.

Philip Newcomb:

Yes, I think a key part of it is that the technology in the tool suites should be highly adaptable. The use of tools that cannot be flexible and adaptable is probably not going to work for you.

Mark Schroeder:

What are the primary goals in Architecture Driven Modernization?

William Ulrich:

The goal is really driven by your business and IT objectives, your strategies and your project requirements. But what you can get to with modernization is the reuse of highly valuable, extremely difficult to replace or rewrite software. You can re-deploy the essence of that software or the really valuable portions of that software, either in large part or part, into new architectures that are going to allow you to really improve how your business takes advantage of the software and improve how you can take that forward. The big key aspect of modernization is the reuse of

business value that's embedded in that software and your data structures.

Philip Newcomb:

... And I would add that, I think that one of the key values that's come from Architecture Driven Modernization, is that it is a dependable, predictable and cost-effective approach to information system improvement and modernization. I think in the past, information modernization has just gotten a bad rap because there used to be a lot of failures. Especially because so much of the process was going manual. As soon as you go manual then you get cost overruns and schedules slide. It can become a real nightmare. So I think that's one of the strengths of Architecture Driven Modernization... is that it has brought about a set of standards, a set of well defined scenarios and tools, technology and service providers that are following that particular approach

Mark Schroeder:

So then overall, it should become a fairly cost-effective approach to modernization versus just throwing money at a problem throughout the course of the project?

Philip Newcomb:

... And risk-free. The last thing you want to do is start a modernization project that fails because you end up with nothing at the end of the day. We have come into any number of projects where they started out by trying to do a manual rewrite for instance.... and two years in they had a mountain of bugs and problems and they just gave up on it. So we were able to come in as the rescuers on any number of projects through highly effective, low risk with very few errors in a fully automated transformational process.

Mark Schroeder:

There are some certain categories that we see in ADM. Could you explain those three categories to me?

William Ulrich:

Sure, and you're probably eluding to some of the breakdown of some of these task categories within the book. The one obvious one is understanding systems. So there is a degree of analysis that would be performed at the front end and ongoing portion of any modernization effort. Just like you don't want a doctor operating without taking x-rays and doing an analysis on what you have. So the other two aspects pretty much go hand in glove. It is important to understand that this whole thing is not a waterfall. The other two aspects are re-factoring and transformation. Transformation gets you into some new architectures and re-

factoring really addresses some of the things in the current architecture that's going to help you get there. So, those two, over the years, those two sets of tasks or category tasks have been intertwined where re-factoring may be pushing out some redundancy or consolidating some redundancy at a basic code level. Where transformation might be moving that whole thing into more of an object oriented view. So those have been highly interwoven in some of the more automated tools and they really need to be looked at as sort of an interwoven set of opportunities in terms of what you are trying to do with your systems.

Philip Newcomb:

Yes, another way of looking at the categories and these overlap with the various types of ADM scenarios and ADM standard packages, is reverse engineering, re-engineering, and forward engineering. So those are the broadest three categories that tend to overlap the whole field. Because you reverse engineer to capture knowledge about systems. You re-engineer the system using various types of re-factoring focused on code level, design level and architecture level.... Re-organization of the application... and forward engineering is the regeneration of a new system from a collection of models. So, along the way and it's very important that the models have extremely high fidelity so that there are no gaps in that process. You have to have complete closure from the source system to the target system in order to achieve the level of automation that is needed so human beings

don't have to come in continuously to make up for deficiencies in the process. So if you do that, then you can have a perfectable process. Its model-based, rule driven from end to end. It's great if you can augment that model-based and rule driven process with something that allows you to provide visualizations of the design. Such as the UML design or other types of support for different design methodologies.

Mark Schroeder:

Talking about different methodologies, ADM is a fairly new method that's come out. What are some of the traditional, historic approaches to modernization and how successful were they and what kind of challenges did companies face with these?

William Ulrich:

I wouldn't call ADM new. I would call it more of a formalization of what's come before. Philip's company has been around for many years. Philip and I have been doing this work for decades. So a lot of what's reflected in the overview sections and the case studies in this book and other work that we've done is really an evolution of the historic approaches that have worked and had been brought forward. So, ADM does not run counterintuitively to maybe other predecessor approaches. I created a methodology in the 90s that was used commercially to do modernization work on a service basis. None of what we have in the book runs counter to

those concepts. I would say that a lot of the work that Philip has done now in the last few years is building on the kinds of things he was doing throughout the 90s.

Philip Newcomb:

Yes, I would absolutely echo what Bill has said. ADM is a name that is given to an organization within the OMG(object management group) and it describes an approach to information system modernization that has evolved since the late 1980s. The primary characteristics of that is that we use models to represent information systems. The modeling is both specific to particular languages and platforms as well as independent. The theory is you take a system, you model it using a grammar system then you transform it into an intermediate language neutral representation. The technical term for that is a platform independent model or a platform-independent language. Then you transform that model into a target specific language and platform. Those are concepts that arose a long time ago and they have stood the test of time. If you look through extensive literature like conference reverse engineering or computer science courses that are taught around the country, you'll see that this is the technique that is advocated by true computer scientists.

Mark Schroeder:

I guess since we're sort of talking about modernization we should look at what are some of the benefits that companies are going to achieve by modernizing their application?

William Ulrich:

One of the key benefits is the ability to realign their systems with their business architecture. So over a window of time your goal is to not close out your systems or certainly not to spend an enormous sum of money trying to successfully or unsuccessfully rewrite something from scratch or even with packages. But more so to take and adapt and align your systems to make or tune what your business is trying to accomplish. So one of the key things there is we're not throwing things out the window, we're not just tossing them away. We are taking what's useful and valuable and reusing it and the process, the approach and the concepts are smart enough to know that if something isn't useful or isn't adaptable, that those pieces can be discarded along the way. So the whole concept is evolutionary adaptability of your current software, to get the most out of it for your business going forward without having to try to throw out the whole thing to try and rewrite it.

Philip Newcomb:

I might add a few things to that. I think that the benefits that our customers are citing for modernization are such things as they are

running on very old hardware, that could be replaced with laptops. And the computers that they were using cost \$25-\$30,000 a pop and could only run at 300 MHz. By transforming the code into modern language they don't have to use the old compilers. For instance ADA compilers on that type of hardware is extremely expensive and they can use basically commodity kinds of IDE's for the future maintenance of their software system and run it on commodity hardware platforms. When system such as that one occur in thousands of units there are an amazing amount of savings on hardware and software costs through modernization. I think other benefits that organizations get if they Web enable Legacy systems is that they take an application that was running on green screens and they extend it to the Web... so that they can allow end users to get direct access to that application. Now the Legacy applications will help be modernized, made into true, multi- tiered, web enabled systems and that makes those systems have a tremendously greater value to their clients. They are more accessible to the client's. Other problems have involved the cost of the licensing. We have another client where we were modernizing a 5.1 million line pro COBOL system to see for them. They were able to pay for the modernization in one year and save licensing costs and that's not atypical. Other things that are motivators for instance, new standards are emerging that are based upon modern software. For instance you'll see that some organizations are looking forward to the future saying that we need to be in J2E for instance, in Java and we need to have these types of component frameworks. And if the application is written in an older language that is not compatible or compliant with the

new standards, then the application needs to be modernized in order to become compliant with new standards. There is any number of reasons that are motivators of that kind but they are major and they effect the entire liability of the business.

Mark Schroeder:

Well as they are doing their modernization like this... Is there a specific architecture that companies should target or do you find that there is a lot of different options for companies based on their business needs?

William Ulrich:

Modernization doesn't restrict the target architectures. Those target architectures change and evolve. For example now we're introducing the idea of cloud computing. Modernization has been around for many many years and you know predates a lot of what you think of today as more traditional current architectures. So modernization does not discriminate in any way shape or form in terms of where you want to go, it will help you get there.

Philip Newcomb:

Yes, I would tend to agree. What we're seeing is that our customers are going to C. Sharp and .Net, C++, GCFS and CORBA. They are going to Java, J2E. Most of the trend tends to be towards object oriented languages. I think that's the trend that we generally see. I think on certainly more than 95% of the systems that we've modernized.

William Ulrich:

We also include in the book a couple of chapters on companies who wanted to go UML, although Philip's right. What he's sharing is the large scale trends. If you want to go to UML or you want to go to one of these other environments or architectural representations, you can do that.

Mark Schroeder:

Earlier you mentioned about re-factoring. What are some of the benefits that a company would get if they actually approached their modernization as a re-factoring project?

William Ulrich:

There are a lot of things in your old code that either shouldn't be there or they should have been there at one point in time but they should have been removed. So if you go, and I'm just going to take this from a business perspective for a minute, because so

many people have changed code over so many years it has gotten convoluted.... and because people didn't understand the code they built in extra code to protect themselves from getting that three o'clock in the morning phone call that the system failed. So there's a lot of things like various tests and there's incomprehensible or certainly not standardized views of the data representations and the code. The list goes on and on. It doesn't matter the language. It does that with a lot of the languages which you see in a lot of the older mainframe systems. There's a lot of code that's very implementation independent and has been left over for decades. So one basic approach is, if you're going to do something with the system and you want to keep it around, you could use re-factoring to take that step. And certainly if you want to expect that this code is going to last for many decades into the future, these are the kind of things you want to think about. You may want to re-factor, transform and then re-factor or you may want to re-factor over a window of time and then transform, or combine the two together. So there are a lot of different ways of approaching it. There needs to be a recognition that the last thing you want to do is take a lot of garbage and throw it into your UML diagrams because once they are in the UML, you will never figure them out. But if you're working with things at the code level you have some advantages because you have all these tools that Philip and I have been talking about so far.

Philip Newcomb:

I would just add to the discussion of re-factoring... That generally when you're moving from a procedural language into an object-oriented language and you're moving from a monolithic application into a multi-tiered application... Re-factoring is a part of the process. You're both transforming the language construct themselves as well as transforming from a procedural representation into an object oriented representation. So, programs are becoming classes, procedures are becoming methods of those classes and along the way if you use semi-automated re-factoring, you're going to need to be restructuring the design of the application, its composition. If you are replacing metal wear, you are going to be re-factoring to eliminate code that worked with obsolete and replaced metal wear that was in the Legacy environment with the code and database access paradigms that are in the modern target that you are heading towards. So, for instance if you hibernate an object relational modeling, that tends to encapsulate all the data accesses into data access objects. Then your code becomes a lot cleaner you get clear separation between the business logic layers, the web-basing and the data access layers. And all that comes as a consequence of re-factoring as a part of the transformational process. Now, on top of that, you can add all kinds of custom re-factorings, which are where the organization has developed a re-factoring plan and a strategy, an architecture for the target system and all those things come back into clutch re-factoring strategies that are implemented during the project. Now, what makes this easy is that those re-factoring strategies are implemented as collections of rules that are applied as a follow on to the code

transformation. It's been situated that there is not a discrete line between them. It is a transformational and a re-factoring process combined together that provides the real power to this type of approach.

Mark Schroeder:

I think in a lot of situations, I've seen people refer to re-factoring as one approach and transformation as another approach.

Philip Newcomb:

That's wrong.. they're combined together. They are both descendents of how far back Bill was talking about this goes. Program transformation goes all the way back to the KBS8 program launch-based software program that focused on model-based code transformation. They were going from requirements to specifications, from specifications down to code. With ADM we're combining all that. We go from code, up to specifications, requirements....from requirements, down to specifications, down to code in a continuous loop. Transformation is integral to re-engineering of applications.

William Ulrich:

Right, and it's important to not think of these things as necessarily two separate things. You can go in and break them down and I

wouldn't want somebody to anticipate from the interview, that if they read the interview, they are now an expert on going in and figure out where to use these or not to use these and mix-and-match some of these things together. What Philip and I might think is intuitive, a whole lot of people just have never even opened up chapter 1 on the topic. So it is important to understand what you're trying to accomplish and how to mix-and-match these things to really get your work done to the degree that you want to accomplish your goals.

Philip Newcomb:

Yeah, I agree with Bill, that also goes back to some of the challenges that people face when they start a modernization project. They are not steeped in all this in the way that people who have been living it for 30 years are. It's sometimes difficult for us to understand how huge that gap is and it's really sad when in effects the decisions of an organization and then they go the wrong direction. We as experts know that they're going the wrong direction and that it's going to cost millions or billions of dollars because they're doing it wrong.

William Ulrich:

I was going to say there's a surprising lack of knowledge. Even at, you know, at cursory understanding of this at a management level where an executive would say, I understand there might be some

different ways to do this that might involve modernization. We need to really discuss that and open up the discussion around that. That level of conversation is not even happening.

Mark Schroeder:

I think a lot of times that happens because there's just not the information. People just don't know about all this information.

Philip Newcomb:

It's really unfortunate. I mean, I really am sad at times when I realize how misguided some organizations are because of their lack of understanding of this discipline. The amount of money that is being wasted because of the failure to take advantage of these types of capabilities is really sad to see. I mean I'm not going to name programs or organizations but I know of organizations that are spending billions and billions of dollars, where we have modernized systems for a couple of million dollars. It just is mind-boggling how they can make such awful mistakes.

Mark Schroeder:

So I guess that points out that it's important for companies to really create a modernization strategy?

Philip Newcomb:

Part of that should be taking a really hard look at the market. Go out, find out what's available, use web-based search to enter the right key words to identify what kinds of companies are out there offering modernization services. Take a good look at them and give them a test ride. See if they can transform your system for you.

William Ulrich:

One more comment on top of that. The lack of even going through some basic analysis of what these companies have in terms of understanding what's there and trying to base a strategy on some knowledge of these current architectures. That's not even there. We're the predominant voices in the marketplace about telling people not to look at their current architectures or ignore their current architectures. There are voices out there that are saying that those voices are costing major organizations tens, hundreds, millions, and billions of dollars a year.

Mark Schroeder:

That's one of the things I'm trying to do with software modernization.com...is have a location where a person can go to find a lot of the different vendors and consultants to do this kind of work.

Philip Newcomb:

That's a great thing. I'm glad you're doing that.

Mark Schroeder:

What are some of the different types of modernization tools that a company is going to come across? They've got their strategy, now they need to take the next step. What are the tools they are going to be looking at?

William Ulrich:

I broadly put them into three categories although any given toolset might have all these things, but that's not always true. One is that they are analysis tools to figure out what you're current systems look like. Almost all the tools regardless of what they end up going ultimately from a modernization perspective have some type of analysis capabilities. They can look at systems, they can map out all the parts and pieces and give you a lot of basic information, a lot of advanced information. Then there's a second category of tool, that I'm not going to try to go through feature perspectives, but there are tools that provide analysis and provide sort of selective pieces of re-factoring transformation. They are not automated tools. They are tools that have to be driven by a person sort of in each phase or step of the game and some of

them do rule extractions, some of them are called slicing. It's sort of a hodgepodge of features and functions out there that are inside those tools. But they're not an automated set that's going to take a system from point A to point B. Then I'll let Philip chime in on this part little more. There are automated technologies that will take you through a combined sort of customized re-factoring and transformation slow with the system as a whole. Those toolsets fall in what I consider to be sort of a separate category because of the degree of automation and transformation they provide.

Philip Newcomb:

My company is a service provider so we use these tools and we adapt them. Our goal is to achieve 100% automation because the kinds of systems that we're dealing with are so large. Right now we're working on two systems of 5 million lines code and the transformation can run for a day or two to transform the entire system. Then you iterate and you adapt the tools to get to what we call a clean compiling link. Then after that we make additional adaptations as the system is going through functional testing process. At the end of the day you've got everything required encoding the tools to transform a 5 million line artifact of code into the target language and target architecture that exactly matches the customer's requirements. The skill sets required to do that adaptation are very high. It's a PhD level. It's tough to find a tool that does that kind of a thing and bring it in-house and do it on your own. I've never seen anyone be successful at it. That's what

we do as a business. There are other kinds of tools that Bill was talking about that can sort of fill in some of the pieces and help here and there, but I think that you really do need someone who understands the completeness. The whole picture in the entire tactical infrastructure to deliver something that is with that level of automation and sophistication.

William Ulrich:

Right, it all depends on what you're trying to accomplish as an end goal.

Philip Newcomb:

Yeah, if you have lesser goals, for instance if you want to do test application analysis and extract you know a model of the code then you don't necessarily need to have that level of sophistication. If you're trying to do an end to end transformation, re-factoring and extensive re-factoring functional code level architectural re-factoring. You've got a multimillion dollar project and you're almost certainly going to be dealing with a fairly sizable team. Both a service provider that is going to provide the tools, expertise and we have accomplished work with any number of major integrators on large projects where they are acting as our interface to the customer and they help out with the testing but were the ones who are actually transforming the information systems.

Mark Schroeder:

What are you looking for when you go out and look for a service provider to help you with a project like this? We'll talk about the end to end transformation.

William Ulrich:

For end to end transformation, you're going to look for somebody who is going to be able to provide the high degree automation and organization. That particular company may engage another third party outsider to do some of your on-site handholding, test support, packaging up of the code to ship out. Maybe building any ancillary links to some other systems and those kind of things. That's one category of provider. You may be looking to do some of this stuff on site as Philip talked about. Now everybody wants to do an automated, end to end transformation of each of their systems so you may be looking for someone who just as basic commercial tools, on-site commercial tools, support and skill sets and enough knowledge as to how to put those together. One of the things that I've noticed, that even with some of the off the shelf tooling which we talked a little bit about, that you can bring in-house. Which will not do the end-to-end, highly automated transformations by the way. Even with some of your service providers out there that use these, they are not as knowledgeable as they should be about some of the techniques for being able to

get the most out of some of those tools. That's also true, I would also say about some of the vendors who sell these tools. You know, they're basically selling a hand saw but they don't know the 18,000 different ways to use the handsaw. They just built the handsaw and sold you a handsaw. Then you're hiring another set of people who are going to try to build you something with the handsaw. So you get a lot of this sort of strange handshaking going on. Sort of the on-site tool, and the tool vendor community and that type of thing. But that's really a separate service category than the highly automated transformation services that are provided and any ancillary support around those.

Philip Newcomb:

There is one other another thing that I'd like to mention. I'm not sure if this is the appropriate place to do it or not, but there is a myth floating around about what is called the Big Bang approach to information system modernization. I think that at times there's kind of a mischaracterization of highly automated transformation of information systems as the Big Bang approach. Then it's cast in a pejorative sense. Again, it's sad to see that happen when the technology is actually highly imperative. What we're doing it is, we transform the system and then we look at the results and we make adaptations to the rules and we transform it again. So what we are able to get up to is a very cyclical process of perfective adaptation of the model and the rules to the customer specific requirements. At the end of the day or maybe at the end of a few

months you push the button and you're transforming the system in its entirety. It's coming out, it's running and you're still making refinements. Bill did I do a good job with that contrast between the Big Bang approach and the iterative refinement?

William Ulrich:

Yes, I would add one point to that. There's ways to work this into current environments. Especially with these companies that have these really, really large scale, highly intertwined sets of systems. You know, somebody said to me the other day when we look at the whole thing together we just give up. There's clean points that you can create a place where you can pull out a chunk and do something with it as Philip was talking about and do whatever you need to do to sort of incrementally move through these things. Chapter 9 talks about trying to deactivate an old system over time, which had many different functions which were all sort of all different business pieces and parts. What we did is just starve the thing of data. We just started selectively starving that piece of code from data while we activated the new pieces. So there are a lot of different techniques and approaches for tackling this and again I'm just sorry to say that people just don't have a sense for it. As Philip was saying, in a lot of the transformations you can grab a chunk and then you go through the process and you get it down to a relatively short window and you can just plop that big chunk back in there....right?

Philip Newcomb:

Yeah, given that, there are two other examples. You know we modernize the European Air Traffic Management system for company called Ellis Corp., Ellis Air Systems. We actually modernized three variations of the same systems because they'd adapted it. There's a French version, a Belgian version and an Australian version. So the first system that we transformed we incorporate all the requirements of the customer and then when the second one came along, it shared a lot of the common components in common with the previous system. So the second transformation of the next system was done in a very, very short period of time. We did the same thing when we did the Patriot Battalion Simulation system for Ratheon. They had three variations and we are now working on I think the fourth variation of the system used in different war theaters. Each time that we've done one of those systems we've been able to leverage the expertise and skills that went into the rule sets and models that we applied to the previous one to do the next one so much faster. When you look at the CMM levels of some of the stuff that were engineering, we'll talk about level V, which is an iterative, perfective, repeatable process. That is what we're seeking for with Architecture Driven Modernization.

Mark Schroeder:

So far it seems that you been talking about fairly large systems. Is there a difference as to how a smaller shop would approach

modernization? Say a company with five or 10 or fewer programmers. Are they going to look at modernization differently than a large company?

William Ulrich:

I wouldn't say it's the shop size. It's the nature of the architecture that they are dealing with. The one thing that you find in a smaller place is, what are their options? If I'm looking at an AS/400 shop and I've dealt with a few, but not large numbers because we normally deal with larger organizations. If you look at an AS/400 shop, a lot of what they have acquired over the years has been commercial, off-the-shelf technology and have built things around that. So if you are looking at a given smaller organization that is making an investment in modernizing everything it's got, they have to look at all the other kinds of things that they might do. If there is an off the shelf solution and a small organization isn't as wound around it's systems, as maybe you know, a really huge multi-million-dollar corporation with lots of different computer systems and lots of different computers. They may be able to go with a slightly different approach to it. So, you know, the smaller guys are at a disadvantage. They are not going to have the same skill sets or even the same financial resources to make the kind of investments that the bigger ones are. I would just say that when you start looking at the bigger shops you have to really look at the nature of the systems and what platform they're sitting on and if they are planning on keeping those platforms or if there is some

desire to go to other platforms or if they really do want to go to model. I mean it's really what their current architecture and architectural goals are as opposed to so much that I'm big and your medium sized so we obviously have different needs.

Philip Newcomb:

I think that what we're seeing is that most of the modernization projects that we're carrying out are being conducted by large organizations. These smaller organizations tend to be extremely risk averse and as tight as things are economically they have a hard time coming up with the kind of money they need to replace their systems. Even if you're down to a dollar to two dollars per line of code. If you're talking about 1 million line system that the organization built over 15 to 20 years. So there's some investment costs in that system. If you just look at the cost that they spent over the years... that represents the value of their company. You know, whatever they could invest they put into that information system and now they need to plunk another million or \$2 million into a system modernization, that's a big chunk of change for most organizations that are running on say \$3-\$5 million a year revenues.

Mark Schroeder:

Now we're going to take a look at a different approach. When we do modernization, I think it's important to have some certain

standards that you're following. How important is this and what type of standards would you set up in a modernization project?

William Ulrich:

Standards really apply to what am I trying to accomplish with my target architecture? I mean do I have some standards that constrain with what that architecture is like and where I want to go and what standards are not being fulfilled with my current architecture perspective? I look at it a little differently. Chapter 15 talks about 15 basic principles of modernization. They're real simple kinds of rules like, don't try to take apart what you don't understand. You know, assess first. Another one is, assess broadly before you assess deep so that you get the lay of the land. You understand what you're trying to deal with. Another one is existing systems that are involved in a project. That project should have some degree of consideration or view of the modernization plan in that project. There are 15 of them in that last chapter and I think it's a real basic set of some guidelines and principles.... if you follow those, they should reasonably keep you within some bounds of not making, you know, a \$50 million mistake or even a \$1 million mistake. So, that's one of the recommendations that we make. I mean there's others when you drop down a level and I'm sure Philip can follow up on this. There's certainly some basic standards and rules you want to follow depending on the kind of techniques and approaches you are using.

Philip Newcomb:

From my perspective the ADM task force, the Architecture Driven Modernization task force of the object management group is providing a tremendous service by defining a specific set of modernization scenarios as well as a specific set of standards and modeling. Which establishes a set of modeling techniques or modeling form in the language or an interchange of models of different kinds that modernization vendors should eventually adhere to. So it's going to take a while before the various organizations that are out there that are working within the ADM task force are able to adapt their technologies to support all those standards. They establish a benchmark for everyone to aspire to. I think that by articulating the things that should be in these types of technologies and tools, that those standards are helping to raise the water level or make the vendors that are developing modernization technologies better.

Mark Schroeder:

When a company is starting a modernization project or as they're working on it, what are the common pitfalls that there they're going to run across?

William Ulrich:

What I've seen is trying to do things manually that make no sense. They may not even make sense to try to do with tooling. I've seen people toil away for a year thinking that they could get some basic set of business rules out of a small number of programs. So that one aspect is trying to do things manually when there's tools out there that can automate so much of what you're trying to accomplish. The second piece, which is probably more important, that goes hand in glove with that, is when we eventually did straighten out this one group of people that were trying to do this work manually and sort of got them on the right path with tooling and sort of the right help and we gave him some processes and techniques and you know, some coaching and training. Ultimately they didn't have a valid target architecture for putting in this type of information and really the solution that they were trying to do was not deployable. So even in the bigger picture, you know, they were struggling away doing the wrong thing incorrectly and when they finally were doing it correctly they still were struggling away because they didn't really understand what they were targeting, what they were trying to do with it. So, know where you're going first and then apply all the right tools and techniques and skill levels and get the help that you need if you need it.

Philip Newcomb:

I have so many horror stories about people doing things the wrong way. I have to be careful about what I say. I remember one organization I talk to. It was a large government organization telling me that, well we're flow charting this the assembler in the system. I said well how are you doing that? They said, well with PowerPoint and I thought my God. Rather than using automation to model the assembler and then and you know a technology that would generate the flowcharts automatically. I asked, well have many people are doing it and they said hundreds. So, it's mind boggling sometimes how much waste there is in projects that are even going on today or recently and knowing that's going on all the time. The other sort of thing that I find amusing is I remember having one discussion about someone who told me that they need to do function point analysis on a large system. I asked, well if you're planning to use automated modernization to transform it, what is the purpose of the function point analysis? They said, well it's to do estimation. So they were planning to do the estimation of the number of manual hours required to rewrite the system, even though they were looking at automated transformation. So the function point of models doesn't even apply and the use of function point analysis is a complete waste of time. Other examples are, they want to do business rule extraction. So they say what are you doing? Well were extracting the business rules. But if you decide you can to do that, then probably the tool you're using, you're going to be lifting up to requirements and you're going to be doing manual development and you've already decided to do a task that isn't going to take advantage of the ability of the system technologies to transform that system. I

mean, I see all kinds of mistakes being made and again they are painful.

William Ulrich:

Your biggest one, that Philip just brought up there is the whole business rule extraction concept. There are tools that support this and techniques.. You better have a really good idea of why you're doing it and your value proposition of doing it and you may be rewriting something manually. I've got to tell you that I've seen more disasters in business rule extractions than you can count. I mean you run into one thing after another with these things. It's like, how many different ways can people make a mistake with this concept? It's an incredible thing.

Mark Schroeder:

We talked about some of the pitfalls that companies can come across. What are some guiding principles that a company can follow to help make their modernization project a success?

William Ulrich:

I think I probably jumped ahead a little bit ahead of this because I started answering earlier, but there some real basic things. If you're doing any type of project that involves some sort of work that's going to either displace, replace, supersede an existing

system or set of systems, you need to start thinking about understanding, at a minimal, understanding what you think you're going to replace. At the next level, understanding that which are replacing in some degree piece out the overall architecture in terms of what you want to accomplish and seriously consider where modernization can help. If you really think that you're going to do some sort of project and you're going to try to extract the knowledge of what this new system is supposed to do out of the hat of business analysts over someone with time. First of all it's going to cost you a lot of money. They're going to miss all kinds of business rules and information about data that's buried inside the current system that would be a perfectly reasonable baseline if you were using it from that aspect and at least jumpstart the thing. But, if there's a lot of value in what you have out there. Yeah, technologically it may be old, but so what? You can get around that with a lot of these tools and techniques that were talking about. The technological age is not necessarily a reason that it can't be modernized. You know it's just like I said there some real basic principles. Have a value proposition of what you're trying to accomplish. I have seen good modernization projects get canceled because the business didn't understand what value is being provided. Get in front of the business early, explain to them how you're going to be delivering the value, what you're going to be delivering. You know, give them some value along the way that they can see with some of the front ends in conjunction with the business processes and then gain the trust of the business so they'll continue to support and fund the work that can deliver the value long-term. Don't try to do these things in a back room and

say I'm going to modernize this thing and I'm going to spring it on the business and boy won't the business be happy? They might be... but engage them early and get their support.

Philip Newcomb:

Something that I would add to this is, the one that really gets my goat, is when I see bad RFP templates being used by organizations. Either it's taking a template that say was given to them by their analyst that is outdated. So, sometimes the famous analyst organizations are actually using RFP templates that guide an organization down the wrong path. That's one problem. So what you see is a set of requirements in the RFP that are already determining that the organization will be doing a lot of unnecessary things and it is not taking advantage the best available technologies and methods that are out there. You know... I used to see that a number years ago when they tended to be more married to the waterfall approach. So you'd see a lot of the requirements coming up had to meet 2167 A or something like that and these are mille standard kinds of RFPs that were completely inapplicable to a modernization scenario.

William Ulrich:

And you know... I think that what we need to do as an organization, maybe the ADMTF itself, is start developing some RFP templates based upon various types of ADMTF scenarios to

help organizations know what their purpose is and then the types of project that they should be structuring and a way which they should be structuring.

Mark Schroeder:

So now by this time everybody listening to this interview is sold on the ADM modernization approach. How do they get a hold of companies that utilize this approach?

William Ulrich:

Well ...you know the Internet is a great source. Again what always shocks me is that people aren't even remotely familiar with some of the basic concepts. So, you know, they don't go out and look for these things. I know that with the ADM @ OMG homepage that we have with the standards group, there is a vendor and technology listing that's out there. It's not complete or totally up-to-date unfortunately. I have traditionally managed a list of vendors off of my archive website where you know you can get some categories of tools and tool vendors that are out there. So, the big thing is that if you go out and do simple Google searches and you're looking for modernization, you're probably going to turn up a couple different companies ranging from highly service oriented to the ones that just want to sell you their product, you know take the product and go. Then you can start to look at some

of the platforms and so forth. You know there's a lot of technology out there and the internet is a good source for a lot of that.

Philip Newcomb:

Well, and another thing that I would recommend highly is, read the book. We're being told by any number of people who have been reading it, that it is an invaluable resource for any organization contemplating a modernization project. The first three chapters outline the various types of modernization scenarios, the various types of technologies and tools that are available and then the 10 case studies in the book go into tremendous amount of depth of giving overviews of the Pro Tools organizations have successfully used on information system modernization projects. So you've got 10 case studies ranging from 30 to 45 pages that are in great depth. A tremendous set of overviews of the different types of techniques and methods. It's the most comprehensive guidebook to information system modernization that has ever been published.

William Ulrich:

... And I would just add to that, a couple of the case studies use more than one tool and you can kind of get a mix-and-match on that to. So, there might have been a supporting tool that was used for a task here and there. There are a lot of tools and tool vendors that are mentioned within the context of that. It covers a good

gammit as Philip was saying. It's covers a real good cross-section of some of the tool vendors out there.

Philip Newcomb:

Yeah, and I'll just add to that... Bill and I are the principal authors but I think we have about 25 co-authors Bill? Who participated in various projects. The other chapter co-authors and you know you see their insights and not just our's in that book. It's just a wonderful cooperative effort on our part of ourselves and our co-authors.

Mark Schroeder:

I have to agree. By reading your book it was one of the better writings about modernization because you have to go look at so many different places on the Internet or read a lot of different books to learn the same information you get from your one book. So, that was a real breath of fresh air to me.

Philip Newcomb:

1hour1min ..has said some really nice things about it as has??? And???. The number of people who have read it are saying good things about it but it's growing all the time.

William Ulrich:

Yeah, there's some great reviews out there.

Mark Schroeder:

Well, I think they're well deserved. We're coming to the end of our interview. What is the best way that people can reach either of you?... because they may want to continue some of the conversation or pick your brain some more. What is a good way for them to reach you?

William Ulrich:

Well, I'll jump in first I guess. My website is www.tsqconsultinginc.com. All my contact information is on there and they are more than welcome to check that out and come find us and they can always Google my name too. That's another way to find me.

Philip Newcomb:

Yeah and I would echo that. We're authors but we're approachable authors. And if you either contact us at our websites, mine is www.tsri.com. That stands for The Software

Revolution Inc. Or you can reach us directly at 425-284-2770 and reach us.

Mark Schroeder:

... And I will have links for both of these sites on the website, on softwaremodernization.com. So if someone didn't get it they can always go there and find your links.

William Ulrich:

Great, we appreciate that..

Mark Schroeder:

Thanks, the both of you, for your time