

Doing Nothing is Sometimes the Costliest Course of Action when Legacy Systems and Applications are Involved

Transcript of a BriefingsDirect podcast on the risks and future drawbacks of not investing in modernization and transformation.

[Listen](#) to the podcast. [Download](#) the podcast. Find it on [iTunes/iPod](#) and [Podcast.com](#). Download the transcript. Learn more. Sponsor: [Hewlett Packard](#).

Dana Gardner: Hi, this is [Dana Gardner](#), principal analyst at [Interarbor Solutions](#), and you're listening to BriefingsDirect.

Today, we present a sponsored podcast discussion on the high, and sometimes underappreciated, cost for many enterprises of doing nothing about aging, monolithic applications. Not making a choice about legacy [mainframe](#) and poorly used applications is, in effect, making a choice not to transform and modernize the applications and their supporting systems.



Not doing anything is a choice to embrace an ongoing cost structure that may well prevent significant new spending for IT innovations. It's a choice to suspend applications on, perhaps, ossified platforms and make their reuse and integration difficult, complex, and costly.

Doing nothing is a choice that, in a recession, hurts companies in multiple ways, because successful transformation is the lifeblood of near and long-term productivity improvements.

Here to help us better understand the perils of continuing to do nothing about aging legacy and mainframe applications, we're joined by four IT transformation experts from [Hewlett-Packard \(HP\)](#). Please join me in welcoming our guests. First, Brad Hipps, product marketer for [Application Lifecycle Management \(ALM\)](#) and Applications Portfolio Software at HP. Welcome, Brad.



Brad Hipps: Thank you.

Gardner: Also, John Pickett from Enterprise Storage and Server Marketing at HP. Hello, John.

John Pickett: Hi. Welcome.

Gardner: [Paul Evans](#), worldwide marketing lead on Applications Transformation at HP. Hello, Paul.

Paul Evans: Hello, Dana.

Gardner: And, Steve Woods, application transformation analyst and distinguished software engineer at [EDS](#), an HP Company. Good to have you with us, Steve.

Steve Woods: Thank you, Dana.

Gardner: Let me start off by going to Paul. The recession has had a number of effects on people, as well as budgets, but I wonder what, in particular, the tight cost structures have had on this notion of tolerating mainframe and legacy applications.

Cost hasn't changed

Evans: Dana, what we're seeing is that the cost of legacy systems and the cost of supporting the mainframe hasn't changed in 12 months. What has changed is the available cash that companies have to spend on IT, as, over time, that cash amount may have either been frozen or is being reduced. That puts even more pressure on the IT department and the CIO in how to spend that money, where to spend that money, and how to ensure alignment between what the business wants to do and where the technology needs to go.



Given the fact that we knew already that only about 10 percent of an IT budget was spent on innovation before, the problem is that that becomes squeezed and squeezed. Our concern is that there is a cost of doing nothing. People eventually end up spending their whole IT budgets on maintenance and upgrades and virtually nothing on innovation.

At a time when competitiveness is needed more than it was a year ago, there has to be a shift in the way we spend our IT dollars and where we spend our IT dollars. That means looking at the legacy software environments and the underpinning infrastructure. It's absolutely a necessity.

Gardner: So, clearly, there is a shift in the economic impetus. I want to go to Steve Woods. As an analyst looking at these issues, what's changed technically in terms of reducing something that may have been a hurdle to overcome for application transformation?

Woods: For years, the biggest hurdle was that most customers would say they didn't really have to make a decision, because the performance wasn't there. The performance reliability wasn't there. That is there now. There is really no excuse not to move because of performance reliability issues.



What's still there, and is changing today, is the ability to look at a legacy source code application. We have the tools now to look at the code and visualize it in ways that are very compelling. That's typically one of the biggest obstacles. If you look at a legacy application and the number of lines of code and number of people that are maintaining it, it's usually obvious that large portions of the application haven't really changed much. There's a lot of library code and that sort of thing.

That's really important. We've been straight with our customers that we have the ability to help them understand a large terrain of code that they might be afraid to move forward. Maybe they simply don't understand it. Maybe the people who originally developed it have moved on, and because nobody really maintains it, they have fear of going in the areas of the system.

Also, what has changed is the growth of architectural components, such as [extract transform and load \(ETL\)](#) tools, data integration tools, and reporting tools. When we look at a large body of, say, 10 million lines of COBOL and we find that three million lines of that code is doing reporting or maybe two million is doing ETL work, we typically suggest they move that asymmetrically to a new platform that does not use handwritten code.

That's really risk of aversion -- doing it very incrementally with low intrusion, and that's also where the best [return on investment \(ROI\)](#) picture can be portrayed. You can incrementally get your ROI, as you move the reports and the data transformation jobs over to the new platform. So, that's really what's changed. These tools have matured so that we have the performance and we also have the tools to help them understand their legacy systems today.

Gardner: Now, one area where economics and technology come together quite well is the hardware. Let's go to John with regards to virtualization and reducing the cost of storage. How has that changed the penalty increase for doing nothing?

Functionality gap

Pickett: Typically, when we take a look at the high end of applications that are going to be moving over and sitting on a legacy system, many times they're sitting on a mainframe platform. With that, one of the things that have changed over the last several years is the functionality gap between what exists in the past 5 or 10 years ago in the mainframe. That gap has not only been closed, but, in some cases, open systems exceed what's available on the mainframe.



So, just from a functionality standpoint, there is certainly plenty of capability there, but to hit on the cost of doing nothing, and implementing what you currently have today is that it's not only the high cost of the platform. As a matter of fact, one of our customers who had moved from a high-end mainframe environment onto an [Integrity Superdome](#), calculated that if you were to take their cost savings and to apply that to playing golf at one of the premier golf places in the world, Pebble Beach, you could golf every day with three friends for 42 years, 10 months, and a couple of days.

It's not only a matter of cost, but it's also factoring in the power and cooling as well. Certainly, what we've seen is that the cost savings that can be applied on the infrastructure side are then applied back into modernizing the application.

Gardner: I suppose the true cost benefits wouldn't be realized until after some sort of a transformation. Back to Paul Evans. Are there any indications from folks who have done this transformation as to how substantial their savings can be?

Evans: There are many documented cases that HP can provide, and, I think, other vendors can provide as well, In terms of looking at their applications and the underpinning infrastructure, as John was talking about, there are so many documented cases that point people in the direction that there are real cost savings to be made here.

There's also a flip side to this. Some research that [McKinsey](#) did earlier in the year [took a sample](#) of 100 companies as they went into the recession. They were brand leadership companies. Coming out of the recession, only 60 of those companies were still in a leadership position. Forty percent of those companies just dropped by the wayside. It doesn't mean they went out of business. Some did. Some got acquired, but others just lost their brand leadership.

That is a huge price to pay. Now, not all of that has to do with application transformation, but we firmly believe that it is so pivotal to improve services and revenue generation opportunities that, in tough times, need to be stronger and stronger.

What we would say to organizations is, "Take a hard look at this, because doing nothing could be absolutely the wrong thing to do. Having a competitive differentiation that you continue to exploit and continue to provide customers with improving level of service is to keep those customers at a tough time, which means they'll be your customers when you come out of the recession."

Gardner: Let's go to Brad. I'm also curious on a strategic level about flexibility and agility, are there prices to be paid that we should be considering in terms of lock in, fragility, or applications that don't easily lend themselves to a wider process.

'Agility' an overused term

Hipps: This term "agility" is the right term to use, but it gets used so often that people tend to forget what it means. The reality of today's modern organization -- and this is contrasted even from 5, certainly 10 years ago -- is that when we look at applications, they are everywhere. There has been an application explosion.



When I started in the applications business, we were working on a handful of applications that organizations had. That was the extent of the application in the business. It was one part of it, but it was not total. Now, in every modern enterprise, applications really are total -- big, small, medium size. They are all over the place.

When we start talking about application transformation and we assign that trend to agility, what we're acknowledging is that for the business to make any change today in the way it does business, in any new market initiative, in any competitive threat it wants to respond to, there is going to be an application -- very likely "applications," plural, that are going to need to be either built or changed to support whatever that new initiative is.

The fact of the matter is that changing or creating the applications to support the business initiative becomes the long pole to realizing whatever it is that initiative is. If that's the case, you begin to say, "Great. What are the things that I can do to shrink that time or shrink that pole that stands between me and getting this initiative realized in the market space?"

From an application transformation perspective, we then take that as a context for everything that's motivating a business with regard to its application. The decisions that you're going to make to transform your applications should all be pointed at and informed by shrinking the amount of time that takes you to turn around and realize some business initiative.

So, in 500 words or less, that's what we're seeking with agility. Following pretty closely behind that, you can begin to see why there is a promise in cloud. It saves me a lot of infrastructural headaches. It's supposed to obviate a lot of the challenges that I have around just standing up the application and getting it ready, let alone having to build the application itself. So I think that is the view of transformation in terms of agility and why we're seeing things like cloud. These other things really start to point the direction to greater agility.

Gardner: It sounds as if there is a penalty to be paid or a risk to be incurred by being locked into the past.

Hips: That's right, and you then take the reverse of that. You say, "Fine. If I want to keep doing things as is, that means that every day or every month that goes by, I add another application or I make bigger my current application pool using older technologies that I know take me longer to make changes in.

In the most dramatic terms, it only gets worse the longer I wait. That pool of data technology only gets bigger and bigger the more changes that I have coming in and the more changes that I'm trying to do. It's almost as though I've got this ball and chain that I've attached to my ankle. I'm just letting that ball part get bigger and bigger. There is a very real agility cost, even setting aside what your competition may be doing.

Gardner: So, the inevitability of transformation goes from a long horizon to a much nearer and dearer issue. Let's go back to Steve Woods of EDS. What are some misconceptions about starting on this journey? Is this really something that's going to highly disrupt an organization or are there steps to do it incrementally? What might hold people back that shouldn't?

More than one path

Woods: I think probably one of the biggest misconceptions is when somebody has a large legacy application written in a second-generation language such as COBOL or perhaps [PL/1](#) and they look at the code and imagine the future with still having handwritten code. But, they imagine maybe it'll be in [Java](#) or [C#](#) or [.Net](#), and they don't pick the next step and say, "If I had to look at the system and rebuild it today, would I do it the same way?" That's what you are doing if you just imagine one path to modernization.

Some of the code they have in their business logic might find their way into some classes in Java and some classes in .Net. What we prefer to do is a functional breakdown on what the code is actually doing functionally and then try to imagine what the options are that we have going forward. Some of that will become handwritten code and some of it will move to those sorts of implementations.

So, we really like to look at what the code is doing and imagine other areas that we could possibly implement those changes in. If we do that, then we have a much better approach to moving them. The worse thing to do -- and a lot of customers have this impression -- is to automatically translate the code from COBOL into Java.

Java and C# are very efficient languages to generate a function point, where there's a measure of functionality. Java takes about eight or ten lines of code. In COBOL, it takes about a 100 lines.

Typically, when you translate automatically from COBOL to Java, you still get pretty much the same amount of code. In actuality, you're taking the maintenance headache and making it even larger by doing this automated translation. So, we prefer to take a much more thoughtful approach, look at what the options are, and put together an incremental modernization strategy.

Gardner: Paul Evans, this really isn't so much pulling the plug on the mainframe, which may give people some shivers. They might not know what to expect over a period of decades or what might happen when they pull the plug.

Evans: We don't profess is that people unplug mainframes. If they want to, they may plug in an HP system in its place. We'd love them to. But, being very pragmatic, which is what we like to be, it's looking at what Steve was talking about. It's looking at the code, at what you want to do from a business process standpoint, and looking at the underlying platform.

It's understanding what quality of service you need to deliver and then understand the options available. In even base technologies like this is, with microprocessors, the power that can be delivered these days means we can do all sorts of things at prices, speed, size, power output, and CO2 emissions that we could only dream of a few years ago. This power enables us to do all sorts of things.

The days where there was this walled-off area in a data center, which no other technology could match, are long gone. Now, the emphasis has been on consolidation and virtualization. There is also a big focus on legacy modernization. CIOs and IT directors, or whatever they might be, do understand there's an awful lot of money spent on maintaining, as Steve said, handwritten legacy code that today run the organization and need to continue to provide these business processes.

Bite-size chunks

There are far faster, cheaper, and better ways to do that, but it has to be something that is planned for. It has to be something that is executed flawlessly. There's a long-term view, but you take bite-sized chunks out of it along the way, so that you get the results you need. You can feed those good results back into the system and then you get an upward spiral of people seeing what is truly possible with today's technologies.

Gardner: John Pickett, are there any other misconceptions or perhaps underappreciated points of information from the enterprise storage and server perspective?

Pickett: Typically, when we see a legacy system, what we hear, in a marketing sense, is that the often high-end -- and I'll just use that as an example -- mainframes could be used as a consolidation factor. What we find is that if you're going to be moving applications or you're modernizing applications onto an open-system environment to take advantage of the full gamut of tools and open system applications that are out there, you're not going to be doing that on a legacy environment. We see that the more efficient way of going down that path is onto an open-standard server platform.

Also, some of the other misconceptions that we see, again in a marketing sense, are that a mainframe is very efficient. However, if you compare that to a high-end HP system, for example, and just take a look at the heat output -- which we know is very important -- there is more heat. The difference in heat between a mainframe and an Integrity Superdome, for example, is enough to power a two-burner gas grill, a Weber grill. So, there's some significant heat there.

On the energy side, we see that the Superdome consumes 42 percent less energy. So, it's a very efficient way of handling the operating-system environment, when you do modernize these applications.

Gardner: Brad Hipps, when we talk about modernizing, we're not just modernizing applications. It's really modernizing the architecture. What benefits, perhaps underappreciated ones, come with that move?

Hipps: I tend to think that in application transformation in most ways they're breaking up and distributing that which was previously self-contained and closed.

Whether you're looking at moving to some sort of mainframe processing to distributed processing, from distributed processing to virtualization, whether you are talking about the application team themselves, which now are some combination of in-house, near-shore, offshore, outsourced sort of a distribution of the teams from sort of the single building to all around the world, certainly the architectures themselves from being these sort of monolithic and fairly brittle things that are now sort of services driven things.

You can look at any one of those trends and you can begin to speak about benefits, whether it's leveraging a better global cost basis or on the architectural side, the fundamental element we're trying to do is to say, "Let's move away from a world in which everything is handcrafted."

Assembly-line model

Let's get much closer to the assembly-line model, where I have a series of preexisting trustworthy components and I know where they are, I know what they do, and my work now becomes really a matter of assembling those. They can take any variety of shapes on my need because of the components I have created.

We're getting back to this idea of lower cost and increased agility. We can only imagine how certain car manufacturers would be doing, if they were handcrafting every car. We moved to the assembly line for a reason, and software typically has lagged what we see in other engineering disciplines. Here we're finally going to catch up. We're finally be going to recognize that we can take an assembly line approach in the creation of application, as well, with all the intended benefits.

Gardner: And, when you standardize the architecture, instead of having to make sure there is a skillset located where the systems are, you can perhaps bring the systems to where the different skills are.

Hips: That's right. You can begin to divorce your resources from the asset that they are creating, and that's another huge thing that we see. And, it's true, whether you're talking about a service or a component of an application or whether you're talking about a test asset. Whatever the case may be, we can envision a series of assets that make an application successful. Now, those can be distributed and geographically divorced from the owners.

Gardner: Where this has been a "nice to have" or "something on the back-burner" activity, we're starting to see a top priority emerge. I've heard of some new [Forrester](#) research that has shown that legacy transformation is becoming the number one priority. Paul, can you offer some more insight on that?

Evans: That's research that we're seeing as well, Dana, and I don't why. Even though I'm one of the most passionate application transformation people around -- I always think, sometimes to my own cost -- the point is that this may not be what organizations want to do.

They turn to the CIO and say, "If we give you \$10 million, what is that you'd really like to do." What they're actually saying is this is what they know they've got to do. So, there is a difference between what they like and what they've got to do.

That goes back to when we started in the current economic situation. The pressure it's bringing to bear on people is that the time is up when people just continue to spend their dollars on maintaining the applications, as Steve and Brad talked about and the infrastructure that John's talked about. They can't just continue to pour money after that.

There has to be a bright point. Someone has got to say, "Stop. This is crazy. There are better ways to do this." What the Forrester research is pointing is that if you go around to a worldwide audience and talk to a thousand people in influential positions, they're now saying, "This is what we have to do, not what we want to do. We're going to do this, and we're going to take time out and we're going to do it properly. We're going to take cost out of what we are doing today and it's not going to come back."

Flipping the ratio

All the things that Steve and Brad have talked about in terms of handwritten code, once we have done it, once we have removed that handwritten code, that code that is too big for what it needs to be in terms to get the job done. Once we have done it once, it's out and it's finished with and then we can start looking at economics that are totally different going forward, where we can actually flip this ratio.

Today, we may spend 80 percent or 90 percent of our IT budget on maintenance, and 10 percent on innovation. What we want to do is flip it. We're not going to flip it in a year or maybe even two, but we have got to take steps. If we don't start taking steps, it will never go away.

Higgs: I've got just one thing to add to that in terms of the aura of inevitability that is coming with the transformation. When you look at IT over the last 30 years, you can see that, fairly consistently, you can pick your time frame, and somewhere in neighborhood of every seven to nine years, there has been sort of an equivalent wave of modernization. The last major one we went through was late '90s or early 2000s, with the combination of [Y2K](#) and [Web 1.0](#). So, sure enough, here we are, right on time with the next wave.

What's interesting is this now number-one priority hasn't reached the stage of inevitability. I look back and think about what organizations in 2003 were still saying, "No, I refuse the web. I refuse the network world. It's not going to happen. It's a passing fancy," and whatever the case maybe. Inasmuch as there were organizations doing that, I suspect they're not around anymore, or they're around much smaller than they were. I do think that's where we are now.

Cloud is reasonably new, but outsourcing is another component where transformation has been around long enough that most people have been able to look it square the eye and figure out, "You know what. There is real benefit here. Yes, there are some things I need to do on my side to realize that benefit. There is no such thing as a free lunch, but there is a real benefit here and it's I am going to suffer, if not next year, then three years from now, if I don't start getting my act together now."

Gardner: John Pickett, are there any messages from the boosters of mainframes that perhaps are no longer factors or are even misleading?

Pickett: There are certainly a couple of those. In the past, the mainframe was thought to be the harbinger of [RAS](#) -- reliability, availability, and serviceability. Many of those features exist on open systems today. It's not something that is dedicated just to the high-end of the mainframe environment. They are out there on systems that are open-system platforms, significantly cheaper. In many cases, the RAS of these systems far exceeds what we'll see on the mainframe.

That's just one piece. Other misconceptions and things that you typically saw historically have been on the mainframe side, such as being able to drive a business-based objective or to be able to prioritize resources for different applications or different groups of users. Well, that's something that has existed for a number of years on the open system side -- things such as backup and recovery and being able to provide very high levels of disaster recovery.

Misleading misconception

A misconception that this is something that can only be done in a mainframe environment, is not only misleading, but also not making the move to an open-system platform, continues to drive IT budget unnecessarily into an infrastructure that could be applied to either the application modernization that we have been talking about here or into the skills -- people resources within the data center.

Gardner: We seem to have a firm handle on the cost benefits over time. Certainly, we have a total cost picture, comparing older systems to the newer systems. Are there more qualitative, or what we might call "soft benefits," in terms of the competitiveness of an organization? Do we have any examples of that?

Evans: What we have to think about is the target audience out there. More and more people have access to technology. We have the generation now coming up that wants it now and wants it off the Web. They are used to using [social networking](#) tools that people have become accustomed to. So, it's one of the soft, squidgy areas as people go through this transformation.

I think that we can put hard dollars -- or pounds or euros -- against this for the moment, the inclusion of [Web 2.0](#) or [Enterprise 2.0](#) capabilities into applications. We have customers who are now trying that, some of it inside the firewall and some of it beyond. One, this can provide a

much richer experience for the user. Secondly, you begin to address an audience that is used to analyzing these things in their day-to-day life anyway.

Why, when they step into the world of the enterprise, do they have to step back 50 years in terms of capability? You just can't imagine that certain things that people require are being done in batch mode anymore. The real-time enterprises are what people now expect and want.

So, as people go through this transformation, not only can they do all the plethora of things we have talked about in terms of handwritten code, mainframes, structure, and [service-oriented architecture \(SOA\)](#), but they can also start taking steps towards how they can really get these applications in line and embed them within an intimate culture.

If they start to take on board some of the newer concepts around cloud to experiment they have to understand that people aren't going to just make this big leap of faith. At the end of the day, it's enterprise apps. We make things, apply things and count things -- and people have got to continue to do that. At the same time, they need to take these pragmatic steps to introduce these newer technologies that really can help them not only retain their current customer base, but attract new customers as well.

Gardner: Paul, when organizations go through this transformation, modernize, and go to open systems, does that translate into some sort of a business benefit, in terms of making that business itself more agile, maybe in a mergers and acquisition sense? Would somebody resist buying a company because they've got a big mainframe as an albatross around its neck?

Fit for purpose

Evans: Definitely, to have your IT fit for purpose, is something that is part of the inherent health of the organization. For organizations whose IT is way behind where it is today, it's definitely part of the health check.

To some degree, if you don't want to get taken over or merged or acquired, maybe you just let your IT sag to where it is today, with mainframes and legacy apps, and nobody would want you. But then, you're back to where we were earlier. You become one of those 40 percent of the companies that disappear off the face of the planet. So, it's a sort of a double-edged sword, you make yourself attractive and you could get merged or acquired. On the other hand, you don't do it and you're going to go out of business. I still think I prefer the former rather than the latter.

Gardner: Let's talk more specifically about what HP is bringing to the table. We've flushed out this issue quite a bit. Is there a long history at HP of modernization?

Evans: There are two things. There is what we have done internally, within the organization in the company. We've had to sort of eat our own dog food, in the sense that there are companies that were merged and companies that were acquired -- HP, Compaq, Digital, EDS, whatever.

It's just not acceptable anymore to run these as totally separate IT organizations. You have to quickly understand how to get this to be an integrated enterprise. It's been well documented what we have done internally, in terms of taking massive amount of cash out of our IT operations, and yet, at the same time, innovating and providing a better service, while reducing our applications portfolio from something like 15,000 to 3,000.

So, all of these things going at the same time, and that has been achieved within HP. Now, you could argue that we don't have mainframes, so maybe it's easier. Maybe that's true, but, at the same time, modernization has been growing, and now we're right up there in the forefront of what organizations need to do to make themselves cost-effective, agile, and flexible, going forward.

Gardner: John Pickett, what about the issue around standards, neutrality, embracing heterogeneity, community and open source? Are these issues that HP has found some benefits from?

Pickett: Without a doubt. When you take a look at the history of what we've been able to do, migrating legacy applications onto an open system platform, we actually have a long history of that. We continue to not only see success, but we're seeing acceleration in those areas.

A couple of drivers that we ended up seeing are really making the case for customers, not only the significant cost savings that we have talked about earlier. So, we're talking 50 percent or 70 percent [total cost of ownership \(TCO\)](#) savings driving from a legacy of mainframe environment over to an HP environment.

Additional savings

In addition to that, you also have the power savings. Simply by moving, the amount of energy that saved is enough to light 80 houses for one year. We've already talked about the heat and the space savings. It's about a third of what you're going to be seeing for a high-end mainframe environment for a similar system from HP with similar capabilities.

Why that's important is because if customers are running out of data-center room and they're looking at increasing their compute capacity, but they don't have room within their data center, it just makes sense to go with a more efficient, more densely packed power system, with less heat and energy than what you'll see on a legacy environment.

Gardner: Brad Hipps, about this issue about of being able to sell from a fairly neutral perspective, based on a solutions value, does that bring something to the table?

Higgs: We alluded earlier to the issue of lock in. If we're going to, as we do, fly under the banner of bringing flexibility and agility to an organization, it's tough to wave that banner without being pretty open in who you're going to play with and where.

Organizations have a very fine eye for what this is going to mean for me not just six months from now, but two years from now, and what it's going to mean to successors in line in the organization. They don't want to be painted into a corner. That's something that HP is very cognizant of, and has been very good about.

This may be a little bit overly optimistic, but you have to be able to check that box. If you're going to make a credible argument to any enterprise IT organization, you have to show your openness and you have to check the box that says we're not going to paint you into a corner.

Gardner: Steve Woods, for those folks who need to get going on this, where do you get started? We mentioned that iterative nature, but there must be perhaps low-hanging fruit, demonstrations of value that then set up a longer record of success.

Woods: Absolutely. What we find with our customers is that there are various levels in the processes of understanding their legacy systems. Often, we find some of them are quite mature and have gone down the road quite a bit. We offer some assessments based upon single applications and also portfolio of applications. We do have a modernization assessment and we do have a portfolio assessment. We also offer a best-shore assessment to ensure that you are using the correct resources.

Often, we find that we walk in, and the customers just don't know anything about what their options are. They haven't done any sort of analysis thus far. In those cases, we offer what we're calling a [Modernization Opportunity Workshop](#).

It's a very quick, usually 4-8 hour, on-site, and it takes about four weeks to deliver the entire package. We use some tools that I have created at HP that look at the clone code within the application. It's very important to understand the patterns of the clone code and have visualizations. We have the visual intelligence tools that very quickly allow us to see inside the system, see the duplicate source code, and provide them with high level cost estimates.

We use a tool called [COCOMO](#) and we use [Monte Carlo simulation](#). We're able very quickly to give them a pretty high-level, 30-page report that indicates the size. Often, size is something that is completely misunderstood. We have been into customers who tell us they have four million lines of code, and we actually count the code as only 400,000 lines of code. So, it's important to start with a stake in the ground and understand exactly where you're at with the size.

We also do functionality composition support to understand that. That's all delivered with very little impact. We know the subject matter experts are very busy, and we try to lessen the impact

of doing that. That's one of the places we can start, when the customer just has some uncertainty and they're not even sure where to start.

Gardner: We've been discussing the high penalties that can come with inaction around applications and legacy systems. We've been talking about how that factors into the economy and the technological shifts around the open systems and other choices that offer a path to agility and multiple-sourcing options.

I want to thank our panelists today for our discussion about the high costs and risks inherent in doing nothing around legacy systems. We've been joined by Brad Hipps, product marketer for Application Lifecycle Management and Applications Portfolio Software at HP. Thank you Brad.

Hipps: Thank you.

Gardner: John Pickett, Enterprise Storage and Server Marketing at HP. Thank you, John.

Pickett: Thank you Dana.

Gardner: Paul Evans, Worldwide Marketing Lead on Applications Transformation at HP. Thank you, Paul.

Evans: Thanks Dana.

Gardner: And Steve Woods, applications transformation analyst and distinguished software engineer at EDS. Thank you Steve.

Woods: Thank you Dana.

Gardner: This is Dana Gardner, principal analyst at Interarbor Solutions. You've been listening to a sponsored BriefingsDirect podcast. Thanks for listening and come back next time.

[Listen](#) to the podcast. [Download](#) the podcast. Find it on [iTunes/iPod](#) and [Podcast.com](#). Download the transcript. Learn more. Sponsor: [Hewlett Packard](#).

Transcript of a BriefingsDirect podcast on the risks and future drawbacks of not investing in modernization and transformation. Copyright Interarbor Solutions, LLC, 2005-2009. All rights reserved.