



**The Group CTO of Royal London
Discusses The “Project Sunrise” Unisys Migration
With MSS International CTO**

An Interview

June 2014

Project Summary

Challenge

Scottish Life is the biggest brand within Royal London, a group encompassing several organizations in the UK life assurance and pensions industry. In 2004 Scottish Life were faced with an upcoming mainframe upgrade that would cost many millions of pounds. They wanted to modernize and reduce its costs at the same time.

Solution

With the help of MSS they evaluated different platform options and settled on HP Itanium/RISC servers running HP-UX Unix. MSS migrated their very large life policy administration system – reputed to be the single largest Linc system in the world – and their legacy pensions system.

Results

It was observed straight away that performance in certain areas, especially the batch work, was substantially improved while the costs had diminished. Royal London estimates a 70% saving in on going costs for their main IT platform. They have a more productive system than before and are well positioned for future development.

Environment Summary

ENVIRONMENT	SOURCE	TARGET
CPU	NX6800	HP Itanium/RISC
Database	DMSII	Oracle
Language	Cobol/Linc	Micro Focus Cobol/PL SQL/C
Data Comm	COMS	MCS!Lite

“We are experiencing a rapid ROI as we have effectively halved the operating costs of our Scottish Life systems. In addition we have dramatically improved performance, increased the quality of service to our customers and implemented a contemporary IT infrastructure that helps us respond faster to the business.”

Interviewer: We have with us **Andy McGarrie (AMG)** Group CTO of Royal London and **Nick Wnewkowski (NW)** - CTO of MSS International. Gentlemen, thank you very much for time today.

So Andy, let's kick off with a first question. Could you take us through the events that led to your decision to consider a migration?

AMG: Yes I can.

Royal London was at the time operating a Unisys mainframe back in 2002/03 and the Scottish Life subsidiary of Royal London was running what I believe to be the biggest Linc application in the world. Nick, can you confirm that was the case at the time?

NW: Yes indeed, there are other sites with more Linc code perhaps in different systems but that was the single largest monolithic Linc system in the world.

AMG: That's right.

NW: Very complex.

AMG: Very complex indeed. So, one of the challenges we had was that the system except the application layer was, in particular, beginning to outgrow the infrastructure it was sat on. The application was particularly processor hungry.

The problem we had was the limitation on two processors on the Unisys mainframe and we had to somehow break that dependency. One of the other aspects in play here was that the application was very single-threaded as was the case with many mainframes and applications at the time.

So it was very difficult to get performance benefits from multi-threading processor activity on the mainframe. So Royal London / Scottish Life was faced with a dilemma because we truly believed that as well as the increasing costs of maintaining what was a massive Unisys mainframe, considering the number of policies it was hosting, we had the risk that the system would basically stop and we wouldn't be able to throw anymore power at it to paper over the cracks.

So that was the initial consideration...

There were two strands to the problem, one was the performance of this massive monolithic, very complex application, and the second was the limitation of the mainframe technology in terms of processing power. So we were looking to re-platform basically to give ourselves some headroom whereby we could leverage on more modern technology, spread the load over more than two processors, and then give us time to decide what we were going to do with the application itself.

As I say, the sub-text was that because of the scale of the mainframe we had to purchase to run the application at the time, this was starting to get very costly, as most people would know in the proprietary world.

So we had escalating costs and reducing performance, which was eventually going to end in a bad situation.

Interviewer: So once you figured out that you needed to do something, and then considered more or less what you needed to do, what was it that led you to decide on the vendor of choice obviously in this case; MSS International and the system integrator?

AMG: The key factor for us in selecting a partner to help us migrate from Unisys was primarily the technology. We wanted to partner with someone who understood the Unisys technology, on the basis that this sort of activity is quite niche. I think there were two key players in the marketplace at the time who were capable and had some kind of track record and experience of migrating from Unisys to another platform. At that time the target platform had not been decided. MSS was in that space and they were evaluated and came out as the leading contender.

Interviewer: Could you outline the process that your organisation went through, from initial project sign-off to the implementation itself...

AMG: There was a vendor selection process which was carried out (and you are testing my memory now) but it was probably carried out over a two to three month period.

The reason that MSS won out in that tender was that MSS could take us to, not necessarily a complete platform of choice, but on to more open technology and had a track record for doing that. The other contender in the market would take us from a proprietary platform to another proprietary platform and that felt like “out of the frying pan and into the fire”.

After selecting MSS we set out another three month period of time which was around the target hardware/software stack. We didn’t really consider any of the smaller players in the market, we thought that was too high risk.

It started off a three way vendor selection between HP, IBM and at the time, SUN. SUN were discounted at the first round, and eventually we selected HP, on the Unix platform running the Itanium chips which at the time was quite “leading edge” but they were tremendously powerful chips which was obviously one of our objectives.

Alongside that (and Nick will remember this) we ran a pilot migration exercise that we called the “proof of concept” which took one of the applications on the Unisys mainframe and effectively piloted a migration using MSS’s tool kit and processes to prove that we could get that application running on a version of the target architecture. It was by no means a productionized infrastructure at that time, but it was to prove that MSS had the capability to port the code across, to port the database across to the target database technology, which was Oracle.

The code was a migration from Linc generated code to a combination of code, which was Oracle, PL SQL, but also we had Cobol applications that we were migrating to Micro Focus Cobol. That process ended up being another three month period by which time we had signed off the technical specification of the target infrastructure which we were jointly building with HP.

We also selected all the various other tools and utilities that were required to use with the HP platform, which directly replaced the utilities we used on the Unisys mainframe. So things like print output viewers, and utilities and file editing, utilities that were clearly proprietary to Unisys. We had to select their alternatives to run on the new platform.

All of that was taking place when the build was in play and we started to do the code conversion with two core work streams. One being the Cobol applications, which were the real old legacy applications and the other being the monolithic Linc application.

In total I believe there was something in the region of 22 million lines of code in the migration and basically we set about a conversion test iteration of those two work streams with an infrastructure work stream running along side of that to build the target infrastructure and all the batch processes and everything else that had to be put in place on the new platform.

NW: I think it’s true to say that in selecting the target platform and the hardware vendor one important thing to mention are the benchmarks that were conducted. Perhaps you could say a little bit more about that?

AMG: Yes, to be honest that was the very technical end, I was more at the application end at the time, - but because of the concerns at the time and nature of application that we were migrating, MSS were put under a lot of pressure to prove that the code and the infrastructure of the environment, if you like (the code along with the database technology etc.) was going to be performant when it landed at the other end. So there was a lot of benchmarking and testing going on with HP - I think we used the HP testing capability over in Germany at some point to test the throughput on Itanium chip technology.

I can't remember the exact details of the result but there was no problem with performance and our criteria for acceptance for all of the migration was that we received no worse performance in terms of batch and online response times etc than we saw on the Unisys. And by definition, we would have extra capacity for growth because we could just put more processors into the machine.

So the comparison was on a like for like basis and even at those early stages it was obvious that we were going to at least match at the performance of the Unisys. And, probably in our back pocket we had some comfort that that performance was going to be exceeded. That is what turned out to be the case.

Almost for free we got a considerable reduction in our batch times. Even just having compared the applications like for like. I can't remember the exact details but it was down from a significant average say from 8 hours down to about 4 or 5 hours.

NW: I think that's correct. In fact the other benefit of the benchmarking exercise was to be able to compare like for like the IBM p servers with the HP Itanium...

Interviewer: Maybe you would like to talk a little bit about the testing scenario that you went through.

AMG: Yes, again the testing of these migrated systems is an interesting challenge. It almost becomes (and I have done migrations on different platforms since and its proved to be the case in those migrations as well) where the testing becomes the biggest single test of that application that a company has ever done.

I would probably include pre "go-live" of the application itself in what I just said because effectively, you are stressing the application to make sure that nothing slips through the gap, that the code is behaving exactly as it should do.

We had two major testing streams: one was the batch system which as much as possible you were snap shotting data on the old platform, taking it through the data conversion process that MSS provided, landing it on the target platform, our infrastructure work stream would build in the batch. The batch processor is around that data, and we were effectively

parallel running the batch over and over again and comparing the outputs. Simply put, that is how we tested the batch.

Then on the online, we had a team of users who knew their way around the online service. They were running through test scenarios that we had built (using what was then the Mercury testing toolset, but now I think its HP, HP took Mercury over).

So they were testing scripts, running transactions and these guys were defining the scripts and the scripts were built into the software, and the software executed the tests. This is very much an iterative process. It's obviously very difficult to estimate: number of tests required to execute, pass rate, acceptance criteria and also duration. But I think there are a number of parameters you can use to decide how difficult or how complex a test needs to be and how much of the code you are going to test and execute before you trust that the conversion has worked.

I can't really say anymore than that, but I think each individual migration will be slightly different to any other previous migration... I'm sure that MSS has found that by going around the world and executing these projects. It's the test plan and strategies that are almost unique to the environment. Providing you have a good understanding of that environment and you have people who understand what the outcomes should look like, then it's a relatively straight-forward parallel runtime test execution.

Interviewer: Nick - from the MSS side can you talk about how MSS consulted and run the testing from your perspective working with the customer as well as HP.

NW: Yes, as you say Andy - every migration is a little different and the aim of migration testing is to ensure that the new application performs as well or better as the old one and, functionally it does the same stuff.

Sometimes customers will have extensive test suites already in place that can be reused and sometimes they have testers which are familiar with the system and can quite confidently say that yes this is working or no, it's not and here's a fault.

We always rely on our customers, and in this case the Scottish Life users were particularly knowledgeable. We approach different migrations as essentially black boxes. We need really good co-operation with our customer's testers and the IT department, and we certainly got it at Scottish Life.

The test team was formed of Scottish Life employees and we contributed several people to that and obviously we solved all of the bugs that came up.

With every large migration there will be faults and the aim is to find them and fix them quickly. I think we managed to keep up a good fix rate with every problem that came up.

AMG: If I can jump in on that quickly regarding the knowledge of the (customer) organisation going through the migration. I have heard debates in the IT world and discussions around these types of projects where people have talked about contracting the supplier, in this case, MSS to test the application and sign-off and prove exactly what the source application does.

Actually, if you think that through - it's a little bit floored in its thinking because MSS doesn't have business knowledge of the application and certainly as we are a financial services organisation we have considerable obligations to meet. We are heavily regulated and have an obligation to sign-off any code that we put into production. There is no way, even if MSS had said that they would do all the testing and would effectively provide a warranty on the migrated code, we would have agreed to that. We would have no comeback really as we can't outsource our regulation.

It's imperative that the organisation whose responsibility it is for the system, the data and the outcomes, is responsible for the testing or proving the testing and signing that off. I think to actually approach a supplier and suggest that they do that, is a little bit far-fetched.

The reason I mention it is because I have had organisations who have asked me about it and talked about the very same thing, and I just don't think it is realistic. They have to have the subject matter expertise and the ability to sign-off the migrated code.

Interviewer: Can you talk us through the timescales associated with the project.

AMG: Our project beginning to end, if you count the beginning as the decision to go with MSS and board approval, was round about March time. The proof of concept work and the selection of the hardware vendor was about another quarter which took us into the middle of the year. Bearing in mind we had this massive Linc application but we also had probably half a dozen other applications to test as well. So we had different test streams testing each application.

When we got into June/July it became one big testing exercise really and a build exercise on the infrastructure space. So the infrastructure guys were building all the target infrastructure, they were testing that, they were operationally accepting that environment, they were building all the utilities in the print mechanisms and everything around the outside of it.

The environment at Scottish Life was particularly complex. I've worked in this space for thirty years, and I have come across a lot of applications: Life and Pensions, Administrative applications and as I quoted recently in a presentation that I did, this application, the Linc application - I have never seen anything so complex in my life. We are talking about putting our hands on about thirty of these things, different types of flavours. And this is massively more complex than anything else.

To the extent where, and one fact that sticks in my mind Nick (and you might remember this) is when we got to interfaces, as in interfaces between the Unisys and other technologies, be that out onto the internet or on to other technology platforms. Most organisations would have a flavour of

FTP or one or two other mechanisms to get on to non-proprietary platforms. In Scottish Life's case they had 36 different types of interfacing. That's not 36 interfaces, that's 36 different methods of interfacing between one computer and another one basically. We had to develop a solution for all of those requirements outside of the migration of the code and the database itself.

So that was all part of the build work and all of that was being tested, alongside what I have described as batch parallel testing and user acceptance testing, and using of the online system. It was an iterative process, and it went through, all in all, in about nine to twelve months. Then the rest of the three months and probably some of that twelve months were spent parallel running the landed application with the Unisys application in a test environment to ensure that all of the results and the outputs were the same.

There was a massive piece of work to prove what took off landing from a data perspective. Again going back to the regulated business aspect, our actuaries were absolutely adamant that every penny in that fund that left for Unisys had to land on the target platform and we had to write quite complex routines to make sure that we had algorithms which could prove that the data had migrated effectively.

All of that was building up in the last three months before go-live and the go-live itself was I think October in the end. Maybe September / October the following year and massive cut over. There were some issues as you would expect in the first month or so but nothing I can remember that was significant in terms of show stopping.

What do you think Nick?

NW: I remember the cut over weekend well. We spent the weekend up in Edinburgh watching the processes go through because it takes a bit of time to cut over all the data, and as you say, the financial data is absolutely crucial. The internal auditors were there checking the results on the Oracle database to make sure they were exactly identical to the DMSII database and they signed off and were happy.

I think they were a few connectivity issues, a few networking issues at the beginning of the week and one or two minor faults, but those were fixed almost instantly.

AMG: I think one of the challenges we have again, and this is a bit financial services specific, but the actuaries wanted something specific in this circumstance. We have done migrations since and this is what actually happens. They run what is called a “valuation”, which values the entire book of business on your system...and that is what the actuaries use year-end and quarter-end to value the business in order to report into the various regulatory bodies.

So their idea of “I’m confident with the data” is to run a “valuation” at the point in time we do the migration on the old platform and run another valuation on the new platform and simply compare the results.

We have used that in recent times on other migrations. However in this case the valuation of this particular system, because of the constraints that we have talked about on Unisys, took five days. So - if we were going to do that, given that it took all weekend to migrate the system and the data, we would have had to effectively shut the business down for five days while the actuaries compared numbers and signed it off. And that clearly wasn’t an option.

So, in this case we had to write these complex algorithms for the likes of internal audit and the actuaries to say “yes I am happy”. This was a fairly complex process. In some cases you wouldn’t need that level of rigour but certainly we needed to. We had to prove every penny, so to speak.

We probably tested quite heavily, but if you were happy to take a more risk based approach, then you might have tested slightly less than we did, but because of that level of testing, the cut over went quite smoothly... Wouldn’t you say Nick?

NW: Yes very smoothly.

We keep going back to the fact that this was a massive application. By the time we had got to go-live, we really had thrashed the systems to death – tested, tested and tested again. I recall there were a few environmental issues but I don’t believe there were any issues of functionality.

AMG: One of the big risks with cut over is that you can never really replicate production load in a test environment.

We used test tools like Loadrunner to try and load the system, to mimic 50, 100, 200 users being on the system. But actually during the early days of the testing, when we did that, we found that we could easily prove that 50 people could use the system. But, when we put 10 real people down, the system fell over. So they are useful tools to an extent, but they don’t really replicate production load because they follow scripted pathways and a normal set of users do not.

One of the concerns from day one is that you've never actually tested out some of the performance issues. But that did not manifest itself at all, which I think is testament to the rigour that was exercised during the testing processes. Both automated testing and real human being testing.

We had these weekends where we would pull in a big set of users and put them on a test system and ask them to work all weekend. Effectively inputting various data to try as much as possible to align the simulation to that Monday morning effect when we went live – and it went very well.

To put this into context - on the same system when we did an Oracle upgrade more recently, I think probably around 2010, we lost the system for a couple of days because the differences between the two Oracle versions caused more difficulties than the actual migration itself. Now that tells us something about the complexity of this application - the fact that an update to Oracle would cause it to fall over and that was tested for six months – it went live on the Monday and fell over on the Tuesday night and there were a couple of days of getting that back.

I would regard this as a massive tick in terms of how the testing system went live.

Interviewer: I know we have already discussed the number of challenges that arose during the project from MSS's side but were there any additional challenges that you would like to talk about Nick?

NW: One of the things we did find was that some of the systems had been designed a long time ago and for a different mainframe. Some of the Cobol systems had originally been written for the old Burroughs mid range systems (the v series) and written in a style that was current at the time but what had happened was that they had used flat files a lot, rather than database.

In many cases the files were not described centrally which is the modern practice. Because of this there was a lot of work to do to match up the file formats. When we talk about database data (and we tend to focus on database data because that's where most of the data is these days) we think of the database being centrally described and very consistent, while the flat file data is not necessarily centrally described and different programs can have different views of it.

AMG: That was it... that was one of the problems.

NW: It was yes. Because Unisys data formats are quite unique in that they allow addressability down to bits and bytes, they don't impose restrictions that you have with signed bits allocated and so on. It can be very tricky getting the data migration to match up.

Aside from that I think most of the issues were in the special bits of the system - in other words the flat file handling, the interfaces, which we've already mentioned. That is one of the things that people miss when they look at a system - they say "how big is it?" - well, its five million lines of code and so on - that can be one of the minor factors in the time scale and effort required if there are a lot of interfaces and those interfaces are very complicated.

All in all, the biggest code base, the Linc code, was perhaps the easiest to migrate, but the most difficult to test, because it was the most complex and the Cobol was smaller but it was substantially more difficult to migrate. It itself had a lot of variety of interfaces, so yes, it was quite an effort to deal with that.

AMG: I would echo that.

I think the more structured the data is, if it's sitting in database management systems, the easier it seems to be to migrate. In terms of the challenges I think MSS had enough experience at the time and probably even more so by now.

What we were able to produce is what is called a "Quirks" document which fundamentally outlined some of the weird things that Scottish Life had done within their systems in the past which MSS had not come across before and had to code in a bespoke way. That was what presented the biggest challenge as it was almost uncharted, unknown. It must be said however, that all of those challenges were overcome.

NW: Some of them were extremely interesting to overcome and we learnt a lot on this project. A lot of what we learned has been integrated into the current tool kit.

AMG: I think you were being kind there when you described some of the older systems as being written in the style of the day... Another way of putting it I think is that you almost had three generations of applications on this mainframe. So as technology had moved forward in the proprietary Unisys world, obviously culminating in Linc type applications, Scottish Life had moved forward with it but had never actually decommissioned any of the older systems.

So you had at least three generations of style of application and that is more of a challenge than the line of code type measure that a lot of people use for this sort of thing, in terms of measuring the scale of the problem.

People ask me about migrations due to the fact that we have had the experience migrating away from Unisys, specifically about how you size the project. I have said that using lines of code as a pure unit of measure in these things is a blunt instrument. Because it's not just that, it's the quirks in the code, it's the styles that have been used by individual organisations, different standards have been applied - there are a whole load of variables in play. I'd trust MSS to help identify what these specific variables are during the, and I forgot to mention this earlier, what do you call it Nick?

NW: The Discovery phase, yes...

AMG: The discovery phase is a key part of the overall project. I would urge anyone who is considering a migration to go through that because that is the point where MSS have the opportunity to have a deep dive into the environment and identify any of these problem areas that need specific solutions.

Interviewer: Can you maybe explain a little bit about the impact that transitioning to the target environment has had on your work force?

AMG: I understand the concern - We had similar concerns because we had a combination of Linc developers and Cobol developers in the application space and a whole raft of Unisys experts in the sort of "operations analyst programmer" space. Operations is operations but there is a learning curve there (it's a like for like really) - they just use different tools to run the batch processes.

So if I separate the two main areas out: one is the "technical programming systems analyst type" on the technical side, and then there's the "operations programmers", the people who know the nuts and bolts of the hardware. We were blessed with a couple of very bright people in this space in the Unisys world and they came across very easily, they retrained very easily. We did need to supplement those with a couple of Unix experts and in particular Oracle experts. We had no Oracle before we migrated, so there we used a combination of retraining people from the proprietary Unisys database world into Oracle and we supplemented them with a couple of experienced Oracle DBA's, and that solved that problem.

System programmers. It depends on the type of system programmer that you've got. I have operated and run several different hardware platforms over the years and still do. I find system programmers to be generally very clever people and they will retrain on any kind of platform that you put in front of them, basically.

On the application side we migrated the Cobol from Unisys Cobol to Micro Focus Cobol and because they had deep knowledge of the application, what it was trying to do and how it was trying to do it, the move from one Cobol to another was no problem. I think if I remember correctly, because that was my work stream, they picked the code on the job in terms of testing the migrated code and in the end they didn't go through the Micro Focus training program. They had a couple of days overview of the utilities, things like file editors and the sort of technology that Micro Focus brings to the party, but they didn't have any training in Cobol and JCL etc because it's a variation on a theme.

On the Linc side, obviously we had Linc programmers - their application was going across to Oracle, so they went through Oracle training and largely speaking they came through it, the main reason being because they had the in-depth knowledge of the application itself. So they are able to gain enough knowledge of the syntax in the target environment to port

across so we didn't have masses of people leaving the organisation and we retrained across the piece. We probably had a couple of drop outs, a small number. The team size was about 100 / 120 at that time. So we had a couple of drop outs who went to other Unisys sites but largely speaking they came across surprisingly easy.

Were they as productive as they had been on the Unisys platform? - We debated that long and hard within the program at a senior level, about what potential hit we would take on productivity and I'm sure we did take a hit on productivity. It's a very difficult thing to measure in the applications development world because there are so many ways to hide lack of productivity... Having been a developer myself, I can say that. Even today and we are talking eight or nine years later, I walk round the building now and I see the same faces that worked on "Project Sunrise" (as the migration project was known). I understand the concern but for us it was never as a big an issue as we thought it would be.

NW: I think its true to say that every migration that we do there are some people who react negatively initially because their first thought is "they are going to fire me and bring in programmers". It's always true that the knowledge of the application of the organisation and the business is far more valuable then just mere knowledge of a particular system programming language and in general, people thrive.

AMG: That is what has happened really. As well as the fear of "what does it mean to me", "am I going to get fired", especially in established IT houses, there is a fear of change generally.

It doesn't matter what you are doing, "if its different it's a threat", so you've always got that to overcome. Our team came through that relatively unscathed and it wasn't as big a problem as we thought it might be.

NW: In fact I seem to recall that there was a little test, something like three months after going live. Some programmers were set a programming task that they reckoned would take a couple of weeks using Linc and surprisingly to them, it took a lot less using the new tools. Partly we think that was down to the fact that the tools had simply better editors and debuggers in the new environment, which tend to aid productivity... As long as you have overcome the initial "getting used to the new language" thing.

AMG: Again looking at the target environment that we are in... Like I say, on the Cobol side it was Cobol to Cobol, and even myself, and I haven't coded for years, can pick up any Cobol pretty much, read it and understand what it's doing. So it's all about that syntax change in that world.

On the Linc to the Oracle side, when you get into the Oracle world, apart from the more technical DBA type individuals and roles, the code developers are using a form of SQL, which is a very simple well-known generic language. The coding language itself isn't going to challenge anyone who is from that background. As Nick said, if you understand the

application, you understand the business and you understand the business processes that this code is automating, that's 90 % up the ladder really.

I've had more difficulty moving people from a mainframe type platform to say, a web type development platform – such as PHP or Java or .Net than we saw in the migration, simply because the application looked and felt broadly the same, it was doing the same thing. So people brought it across more easily. My success rates of taking people off mainframe type technology onto a completely new one is less successful.

Interviewer: Of major concern for our customers is return on investment. Can you talk to us a little bit about the return of investment scenario for this particular project.

AMG: Like most organisations, we're not very good at benefit realisation - so we tend to take a long time to make a business case, to move forward with something, particularly if it's a significant investment on behalf of our business. We get the go ahead, we pile into the project, we deliver the project, we move onto something else and we never go back and say "how much did it save or how much did it cost us?"

Obviously we were tracking costs in the project itself so I can't put a figure on that but, what I can say is that for the hardware platform alone, we were paying Unisys a significant chunk of money - into the millions. That went back into the early Noughties, and we were faced with pretty much every four to five years, a capital hit of 4 to 5 million pounds because we had to replace what were becoming quite sizable mainframes because of the processing constraints I talked about earlier.

I think the production machines we were using were 500 MIPS Unisys mainframes - basically used to run one application. Which is a big beast in that context. That was the cost profile we faced at the time which was becoming unmanageable given the size of the book of business, the revenue and everything else.

I think towards the end when we threatened to migrate, Unisys didn't change their cost model at all. When we got serious and they believed we were going to leave them, they started coming at us with all kinds of attractive offers that would dissuade us from making the transition. At the end of the day it became a risk call more than a financial call. I truly believe even to this day that had we still been on the Unisys mainframe, that application would have stopped by now and we would have been stopped from trading. That would have been really serious.

Having said all that I run the infrastructure now and the cost of the HP Unix, Oracle platform that that runs on is minimal. It's less than half a million pounds. You're talking nearly 10 years ago it was over 1 – 2 million pounds just on a BAU basis and then a significant capital hit every 5 years - so you can work out the numbers yourself. Over the years we would have sunk a considerable amount of money with the Unisys and on this new platform, whilst we have had to invest in it over the years to increase the capacity slightly, obviously we have had to replace hardware as it goes out of support - it's a much lower cost platform and

therefore the investment requirement is not as challenging. So we would have realised a material cost saving over the 9 years since we migrated. I can't put an exact number on it but it is massively material. And every year we save more and more ...

Interviewer: Give or take, including the cost of migration when did you more or less, achieve break even?

AMG: We didn't track this closely enough, even the accountants didn't keep an eye on it, once it's spent, it's spent type of thing. I would say somewhere between two and three years.

Interviewer: We are coming to end of our time together and I just wanted to ask one final question. What would your key message and advice be to any customer wondering what to do next with their legacy estate?

AMG: I think the term "legacy" is over-used these days I have to say. I have responsibility for applications that are no more than two years old and I've called them legacy.

But really there comes a time when you are talking about older applications written on proprietary, mainly mainframe platforms, but not exclusively mainframe platforms - when you have got to do something with them. If you are one of the big integrators, they call it modernization – that's not a bad description.

I think most people need to understand that you can't just carry on doing nothing for ten years and another ten years and another ten years. You've got to make a call and there are several options available to you in terms of getting off a platform and getting on to a different one. Migrational porting of this type is one of those options.

Ten years ago we were very sceptical about it, we thought there were proprietary constraints with these older platforms, it just didn't seem possible to convert a system and run it on a completely different platform all together. That is clearly not the case anymore. The technology and the process is proven.

As I say, at Royal London we have been through two or three of these projects - off different platforms and onto different platforms, and I can vouch for the process.

Is there some pain associated with the process? - of course there is. Every project comes with disruption and a bit of pain. What do you get at the end of it? – well, actually you get what you started with and in this case that's a benefit because that is what you are aiming to do. But that provides you with a launch pad for the future, so currently Scottish Life business is the most productive and profitable business in the Royal London group, even today. In particular this has very much been the case over the last two to three years.

Had we stayed where we were the system would have stopped. It would have been a closed business by now! I think that in itself is a testament.

So, you've got to progress, you've got to move on - you have to look at where you want to be in five ten years time, and you have to make quite brave decisions. We took what we thought was a brave decision and it was a brave decision. It turned out to be the right decision.

Since the application and infrastructure has landed, nobody in our organisation has ever questioned it. When we went back and asked if we did the right thing nobody ever questioned it.

Nick as you know from the whole approval process upfront, particularly for this sort of investment that we were looking for, which is quite unusual for our size of organisation – we were challenged quite heavily on this. In fact Maurice (MSS CEO) was put in front of me and an all-powerful steering group and quizzed quite intensely a number of times before it was all approved. But as I say, retrospectively nobody asked any questions.

If you have got concerns about your legacy you have got to do something about it and this is a viable option.

Interviewer: Thank you Andy, do you have anything to add to that Nick?

NW: I recall very well the approval process and the initial scepticism. In a sense that is quite right, you're a big organisation, you don't take on a big challenge like this lightly. But as we progressed we kept on delivering more pieces of the puzzle, especially the benchmarks. It was a valid question to ask "will these machines that cost a fraction of a mainframe... will they *really* deliver the performance we require?"

Once that was proven, the whole organisation kind of breathed a sigh of relief and said "this is going to work". It's a case of building confidence little by little as you go through.

AMG: In today's world we can be less bothered about performance because of the power of some of these machines. I mean the power available to you these days is such that you never stretch the hardware technology. And certainly some of the older technology can't really deliver the same sort of "bang for your buck" really.

To echo what Nick said - I was a sceptic. I was a traditional mainframe developer. How could someone take that code and convert it to something else - surely that's rocket science.

Obviously, having been through the process, as I say, it's now proven.