
Software Migrations: Past, Present & Future

Author: Mark Wilson

About the author

Mark Wilson is the Technical Director at RSM Partners, a UK based consultancy company. Mark has almost 30 year's involvement in the IBM market place.

Mark spends the majority of his time working with RSM Partners and IBM's customers performing technical services & consultancy in a myriad of capacities from Systems Engineer to Project Manager.

Introduction

"Software Migration" is a term the majority of IT professionals are familiar with and those in the System z (mainframe) market place are no exception. We typically encounter a migration when senior management utter the words "Reduce Costs". However, there are several other reasons migrations are considered:

- Mergers & Acquisitions
- Outsourcing
- Globalisation

As a company changes and the functionality and technology of products evolve, a software solution installed as recently as two years ago may no longer be the best or most cost effective.

Costs are still an issue for system z users even given all the work that IBM has done to help reduce costs. A quick search of the Internet and you find:

For four years running, attendees at the Gartner conference listed third-party software costs as the No. 1 inhibitor to mainframe growth, according to Gartner analyst Mike Chuba.

"This is the tail that wags the dog on mainframe procurements," Chuba said. "In many situations, customers make decisions on what hardware they're going to buy based on their existing software contracts."

38% of the mainframers surveyed said third-party software costs were the No. 1 inhibitor to mainframe growth in their organizations.

Source: SearchDataCentre.com

So, cost is typically the No. 1 driver for performing a software migration. However, beware it's not always the most obvious vendors that cause the biggest issues!

Retaining the existing product *may* risk:

- Higher licence charges
- Increased maintenance overhead
- Vendor support erosion
- Reduced competitive advantage
- Compromised requirements
- Inability to exploit new technology
- Duplication in functionality

Even when a clear cost case exists for performing a migration, the perceived pain and effort required to migrate to a different solution often dissuades companies from doing so.

But its not all bad! Organisations all over the world have totally transformed their software portfolio making significant cost savings and driven new working practices to improve efficiency.

Software Vendor Rationalisation

Many organisations have looked at their procurement processes around mainframe software and found that they are dealing with far too many vendors. Having a vast portfolio of products and vendors carries an overhead that can be summarised as:

- Management overhead
 - Contract Management
 - Licence Management
- Technical costs
 - Installation time
 - Training Costs
 - Support costs

This has led many to review their vendors and, where technically possible, consolidate them. This has meant some products being removed and others migrated. In the past, the attitude was always only if we have a near perfect fit. However, I believe this has changed in recent years as many organisations start to accept less functionality if the price is right.

The goal here is to reduce the number of vendors and then agree a better financial deal with those left in the aim of reducing mainframe software costs. The two big winners are likely to be IBM and CA who are both willing to negotiate. I have seen many cases in the last few years where they have worked with their customers to reduce costs, whilst protecting the investment they have made in mainframe technology.

Some of the issues with software reviews and migrations

Largely driven by the software vendor rationalisation point above, I feel mainframe IT teams will be asked to make some very hard decisions in 2010, just as they were in 2009.

I have worked on one project recently where we were told “Migrate from ISV X to ISV Y”, without any discussion or debate. The costs savings were so great, the senior management team made the decision without any technical review. The edict from on high was “Just make it happen”. I believe our mainframe technical teams will be faced with many such challenges in 2010.

It’s always good to look at what software we have installed and ask ourselves some very simple questions:

- Do we really need it?
- Is there a cheaper alternative out there?
- Do we use the software efficiently?
 - Do we need to run on all LPARS?
- Can we do things differently?
 - Just because we have always done something doesn’t mean its right
- Have we got any duplicate functionality?
- Are there functions in base zOS that we can use to replace functions in ISV products?

So what is a Software Migration?

Simple really! Remove Product A and replace with Product B. Sounds simple? Well in some cases it is.

Software migrations fall into the following areas:

- Key Infrastructure
 - Security
 - Scheduling
 - Automation
- Databases & Application
- Tools & Utilities

Within these categories the migrations themselves are categorised as:

- Complex
 - Typically when product database(s) or application code require change and extensive user testing is required
- Moderate
 - Typically involves end user testing, but is a fairly simple technical process

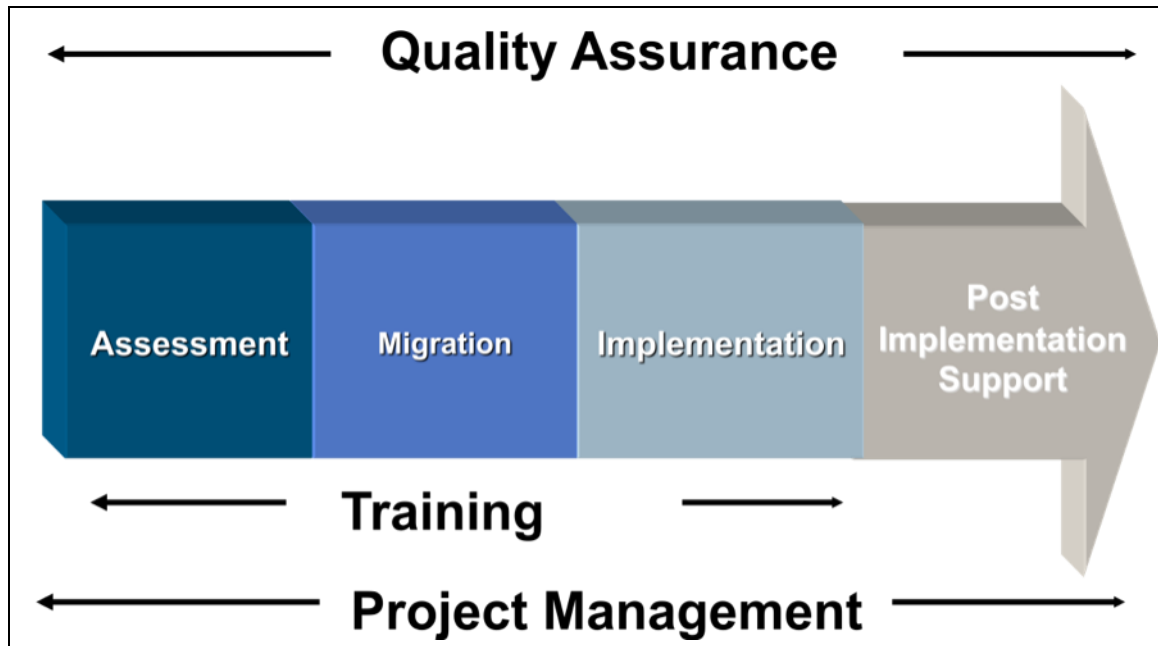
-
- Simple
 - Typically a simple tool or utility replacement that require minor training and can usually be completed with little or no business impact

Process is Key

Process is key when undertaking a software migration and the following steps form the majority of software migration projects:

- Business Case Analysis:
 - Current Software Costs
 - Migration Effort Costs
 - Proposed New Software Costs
 - Prepare ROI Statement
- Migration Assessment
 - Issues
 - Risks
 - Detailed Cost Analysis
 - Draft Project Plan
- Project Planning
 - Project Definition Workshop
 - Project Definition Report
 - Project Plan
- Preparation
 - New Software Installed & Customised
 - Conversion Tools Installed & Customised
 - Initial Conversion Performed
 - Conversion Process created
- Testing
 - Migration Performed
 - System Testing
 - End User Testing
 - Production Testing
- Production Cutover
 - Product Implemented Into Production
 - May need to perform IPL for certain Products
 - Final Data Conversion, if required
- Post Implementation Support
 - Onsite Support
 - Remote Support

Add to the above Quality Assurance & Training and you have your typical Software Migration.



Software Migrations in the past?

In the past software migrations have only been undertaken after a detailed technical review has been undertaken. No stone would be left unturned to ensure that all the nuances of the current product were understood. The technical teams involved would want to ensure that every function currently utilised could be replaced or in some way emulated by the proposed new product.

The Technical team would have a great deal of influence in the decision making process and in certain cases scupper any potential migration, effectively preventing any possible cost savings.

What Software Migrations are commonly done?

There is a vast array of software products on the market from IBM, CA, BMC, ASG and many others. Most vendors have product functionality provided by one of their competitors, so any list can be interpreted many ways.

Detailed below are some of the software migrations I have been personally involved in:

<i>Source Product</i>	<i>Target Product</i>
Zara, CA-1 & CA-TLMS	DFSMSrmm
CA-SORT & Syncsort	DFSORT
CA-Scheduler, CA-7, Jobtrac TWS & Control-M	Control-M or TWS
SAS/MXG/MICS	TDS & TCR
CA-Easytrieve	Cobol

Automation Tools (OPSMVS, AF/Oper,OPSMVS,etc)	SAO
CA-Top Secret & CA-ACF2	Security Server

Migration Toolsets

For certain migrations, specific toolsets have been developed to ease the migration.

These tools are typically deployed where a database of information has to be migrated:

- Security
- Printing
- Scheduling

Software Migrations in the future?

Given the current financial climate some of the technical requirements often suggested by our IT teams may well be excluded from the decision making process as the mantra of senior management is “Cut Cost and just get it done”. This has some very real issues for the technical teams who are faced with modifying newly acquired software products to meet their requirements. This has further resulted to some technicians dusting off the old Assembler manuals and trying to understand what “**OA6B**” actually means.

Summary

Software migrations can be challenging, however, with enough planning/time and the correct skills anything is achievable. But in today’s IT department compromise will normally be required.

The views expressed in this article are solely those of the author.

Mark Wilson can be contacted at markw@rsmpartners.com

* **OA6B** = SVC 107 aka Modeswitch from Problem to Supervisor state or vice versa