



**“Rewriting vs. Migration of
Typical Unisys Applications”**

White Paper

by

**Nick Wnekowski
MSS International CTO**

Introduction

When we consider modernisation alternatives for a typical Unisys application (mixed online and batch system) we have to consider the advantages and disadvantages from the customer's perspective.

The option of replacement with a package is outside the scope of this document. Suffice it to say that its great advantage – that it can add new features and facilities to the application while also changing the way existing functions are handled, “presumably for the better” – is also its great disadvantage. We know clients who have taken on too much change, the package implementation has failed and enormous costs written off.

The straightforward way of renewing and cost-reducing an application is to rewrite it in a more modern language for a more advanced and lower-cost platform. This can be done using specialist tools from a migrations company or simply by rewriting the application in the new language. The latter will require considerable resources and, unless the application is very small, will involve contracting out the work to an outside company.

In the case of Linc, it is possible to move to Unix or Windows and then to Agility Business Suite. Although Unisys offer a “safe harbour” “guarantee” on the second move, it is the first that is fraught with difficulties and is a true migration rather than, as is sometimes presented, a simple move maintaining the same language on another platform. The language may (or may not – there are significant de-implementations to worry about) be the same, but the platform is totally different. Differences in database technology and character set mean that logic errors and performance issues **will** appear. This should be treated as a migration like any other – but, perhaps with less automated tools available to help.

Rewriting

The information we have from Oracle is that they could rewrite a system in Forms with no GUI enhancements in the following timescales:

<u>category of screen</u>	<u>time to write</u>	<u>time to test</u>
Simple	0.5 day	0.5 day
medium	2 day	2 day
complex	5 day	3 day

In a typical system of 500 screens, you might find 50 simple screens (menus etc.), 250 medium-scale screens and 200 complex screens. On the basis of the above, this would take 1525 man-days to rewrite, and 1125 days to test. If a company calculates a cost of £300 per day for an average programmer working on-site (offshore development will be cheaper but will involve higher risk and a large effort to manage and control) this is going to be £795,000.

The cost of rewriting the batch component will be at least as much again, say another £795,000. This would give a total of £1,590,000.

An MSS migration of this system would be around £200,000 based on a code migration only (MSS normally quotes for the entire project including data migration, job control migration, environmental software replacement and management of the project). To this the customer should add some time to perform testing, say 200 man-days (= £60,000). This testing, however, is not something to be billed to the migration only since the customer has to do some work in order to become familiar with the new environment. The cost of rewriting, in this case comes to 6x the cost of migrating.

Is a Re-write possible?

However the above assumes that a rewrite is possible, i.e. the customer system is well enough documented (including all revisions over the life of the system) to be able to be rewritten. In the usual situation for a mature system, a lot of time and effort would have to be invested in understanding the existing code before a rewrite could be attempted. This would undoubtedly push the cost much higher, lengthen the timescale even further and increase the risk attached to such a project.

The other major consideration that might rule out a rewrite is the time involved. To be able to manage a complex manual replacement of a system is hard enough without having to chase a moving target. Therefore the system normally has to be frozen for the duration of the project. Even so there will be vital maintenance fixes that have to be implemented during the project and this will give significant problems in a manual rewrite and extend the timescale significantly and unpredictably.

In contrast, an automated migration can tolerate normal development in parallel with the migration project (as long as the testing programme is designed to cope with this). New code releases can be re-migrated in a matter of hours and new code modules of arbitrary size can be accepted into the migration base without impacting migration times.

Introducing GUI

Having spoken to a contact in a leading British Software company, the cost of rewriting a system using GUI technology such as PowerBuilder, Delphi or Oracle Forms would be much more. Although he would not release any of the metrics they use for estimating projects, he said that implementing GUI technology is much more expensive than for traditional methods. The reason is that the system must be redesigned from the beginning, normally going right back to the business principles behind the application. The resulting system will look very different from the original, probably having many fewer screens but making full use of GUI techniques - pop-up windows, lists of values etc.

Because of the GUI techniques the testing will have to be very thorough and will be time-consuming - there is likely to be no one to one correspondence with the old system that can be used to generate test input and test results on the old system to be compared on the new system.

Additionally, retraining of staff becomes a major issue. In a large organisation where staff are not located centrally, this becomes even worse. Another contact cites the Norwich and Peterborough Building Society experience where a department was formed to do the training. Much of their time was spent just to schedule moving staff around the branch network so as to free up staff to attend training sessions on the new system. With a migration or a rewrite using conventional techniques, this can be kept to a minimum. With a GUI rewrite, this might add another £250,000 to the total.

Testing

With an automated migration there is a one-to-one correspondence between the original screens/reports and the new. If GUI screen features are introduced it will be on a rule basis. Therefore we can create an absolute standard to test against –scenarios run in a testing environment on the original platform can be repeated in the new environment. Using test capture tools the testing process can be largely automated and tests can produce unambiguous success or failure results.

When a rewrite is undertaken, unless there are stringent controls to insist that the replacement code and screen / report layouts are kept identical, the testing process cannot be automated or unambiguous.

Conclusion

Making a guess, based on conversations with various colleagues and contacts, I would put the GUI rewrite of a 500x500 system up to £2,000,000.

Overall, this will be between 8 and 12 times as expensive as an automated MSS migrate! migration.

It will also take far longer and be unpredictable in duration.

It will impose a code freeze.

It will give reduced quality – the code written will be not be as consistent as that generated by an automated tool.

It will introduce higher risk – according to Gartner and others the majority of (manual) software projects of this scale fail.



MSS International GmbH
Bahnhofstrasse 7
CH-7250 Klosters Switzerland
Phone +41 81 420 23 58 Email: info@mssint.com
Web: www.mssint.com
