Fossilization vs. Transformation

Speaker Notes included
with this Presentation

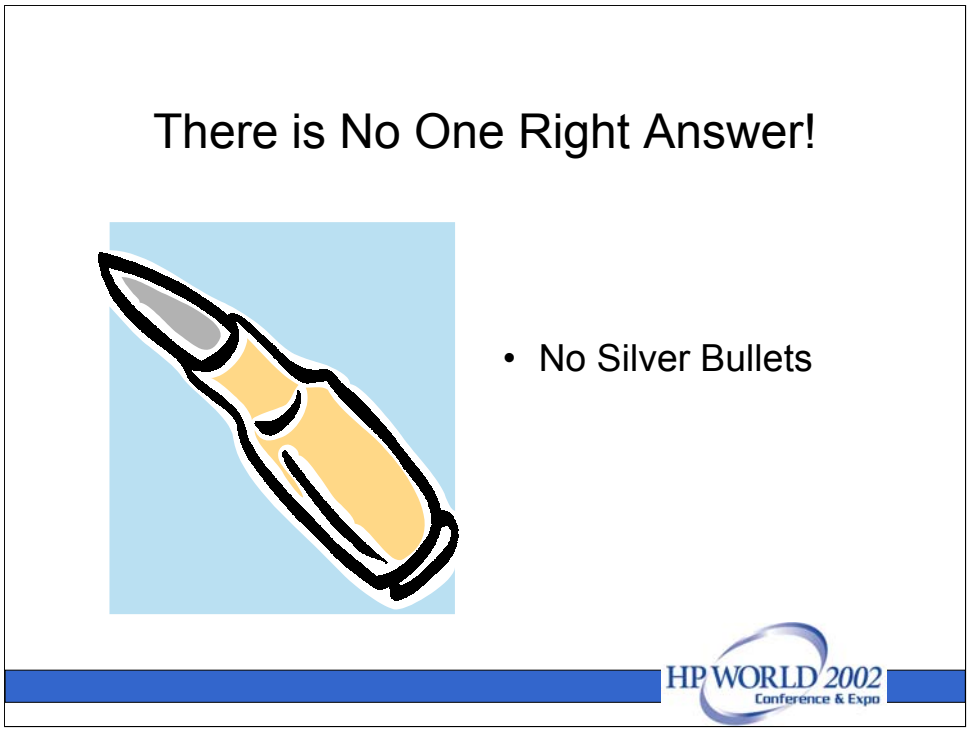
Alan Yeo
CEO ScreenJet Ltd
alanyeo@screenjet.com

About The Author

Alan Yeo is CEO of ScreenJet Ltd a UK based company set up in 1999 to co-ordinate the design and development of the ScreenJet Emulation and Graphical User Interface products for the HP3000.

ScreenJet have also developed migration tools to convert VPLUS forms into ACUCOBOL-GT AcuBench GUI formats.

Alan has also been the CEO of Affirm Ltd since 1986. Affirm are an HP3000 ISV specialising in manufacturing applications on the HP3000. Alan has been designing, developing and implementing VPLUS based applications since 1980.

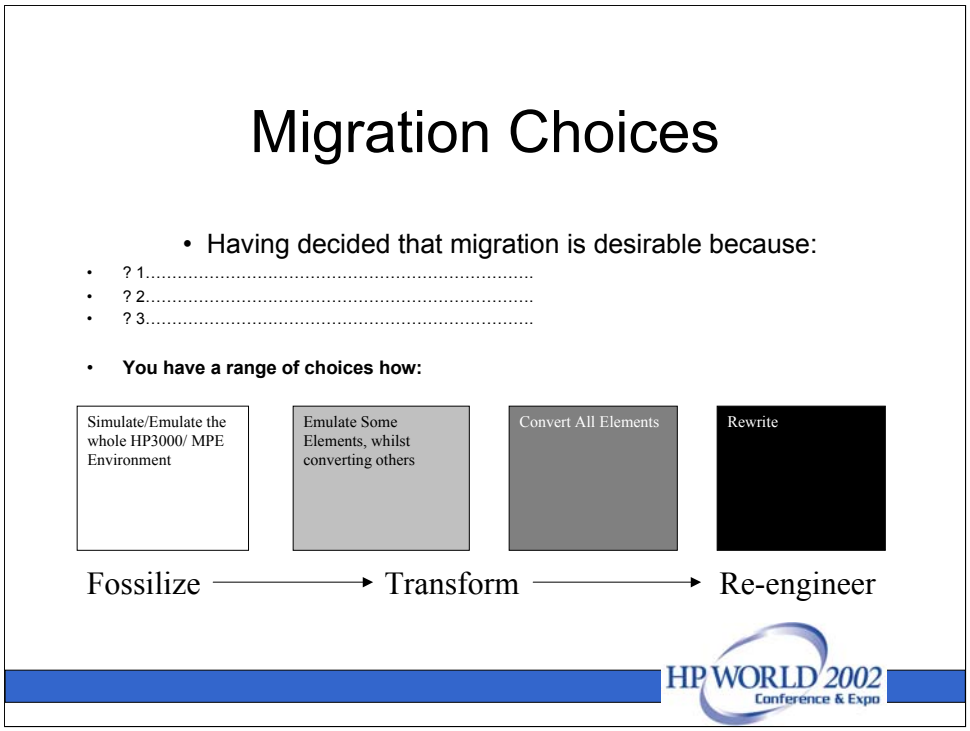


Another of the many migration sessions at this conference, covering the large number of issues faced by organisations contemplating a future without the HP3000.

The fact that there are so many sessions indicates some of the complexity of the situation. "There is No One Right Answer!" There is just a whole range of options that you will have to assess. In fact for many it will be an iterative process where what appears to be a logical first choice of direction may have to be later revised because subsequent detailed investigation throws up technical, cost or time issues that are unacceptable to your organisation.

In this session we are not going to try and answer, "should I migrate some or all of my HP3000 applications" but assuming that you think that you do, to look at the main types of migration possible, and the upsides and downsides to each of them.

Of course at the end of the process you may well re-asses your whole strategy and decide that migration is out, and decide to either buy a warehouse full of HP3000 kit to keep you going for the next fifteen years, or to junk the whole lot and start from scratch.



Having decided that migration of an application(s) is initially desirable because:

? 1…….

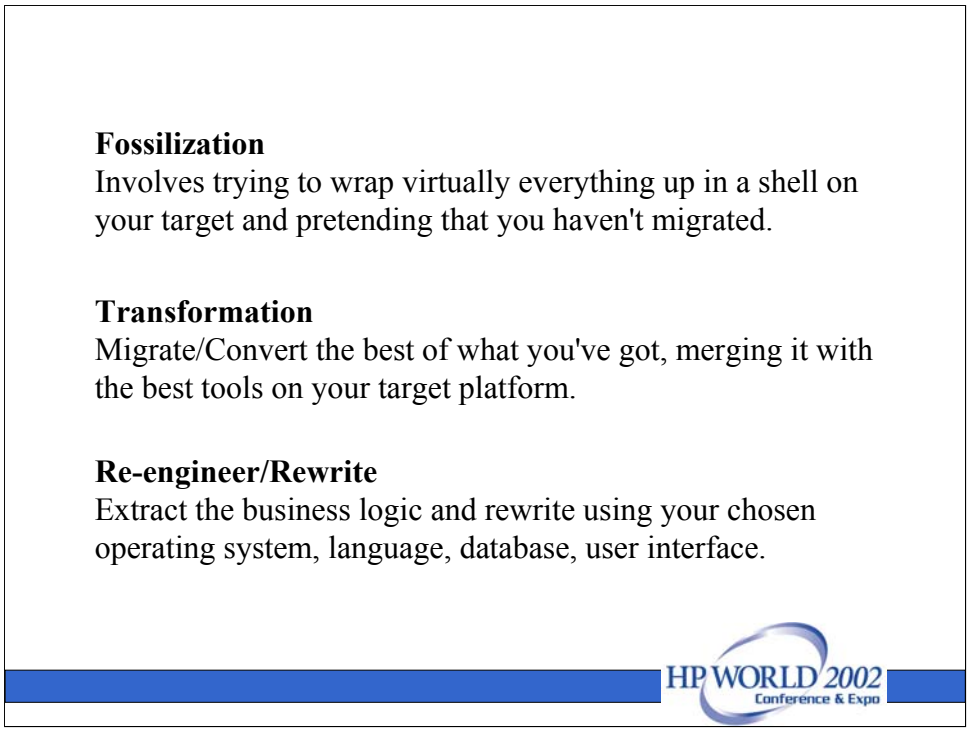
? 2…….

? 3…….

You have a range of choices how:

Fossilize Transform Re-engineer

It would be nice if there were only four choices to make, however in reality the tools and services currently available mean that there are very many shades of grey between the Emulation and Rewrite options. I have chosen the following definitions:



Fossilization

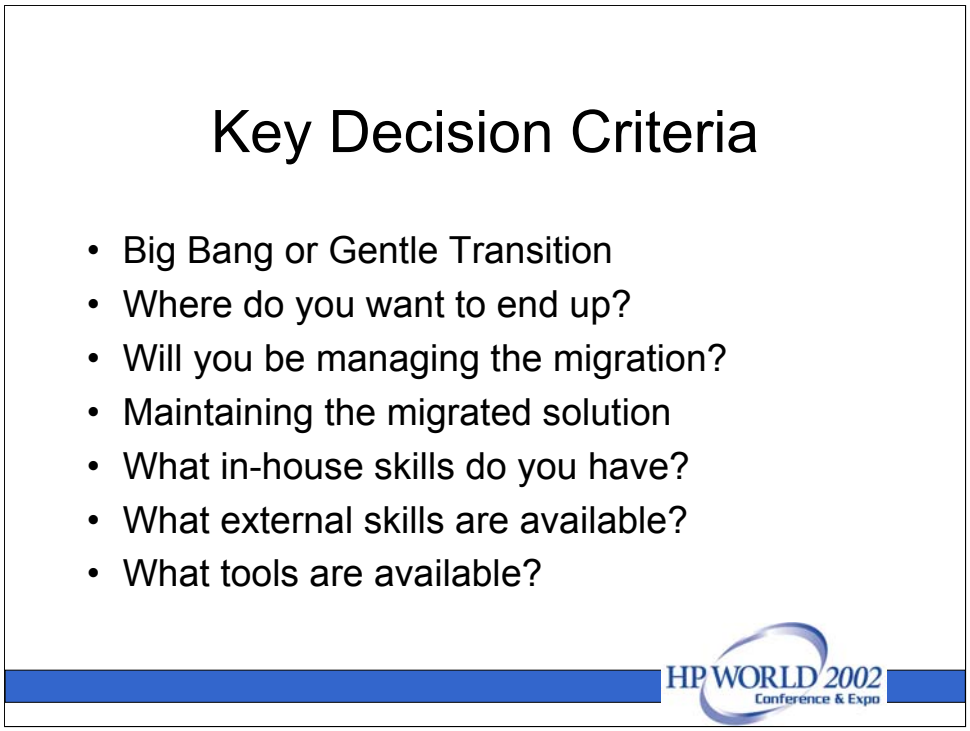
Involves trying to wrap virtually everything up in a shell on your target and pretending that you haven't migrated.

Transformation

Migrate/Convert the best of what you've got, merging it with the best tools on your target platform.

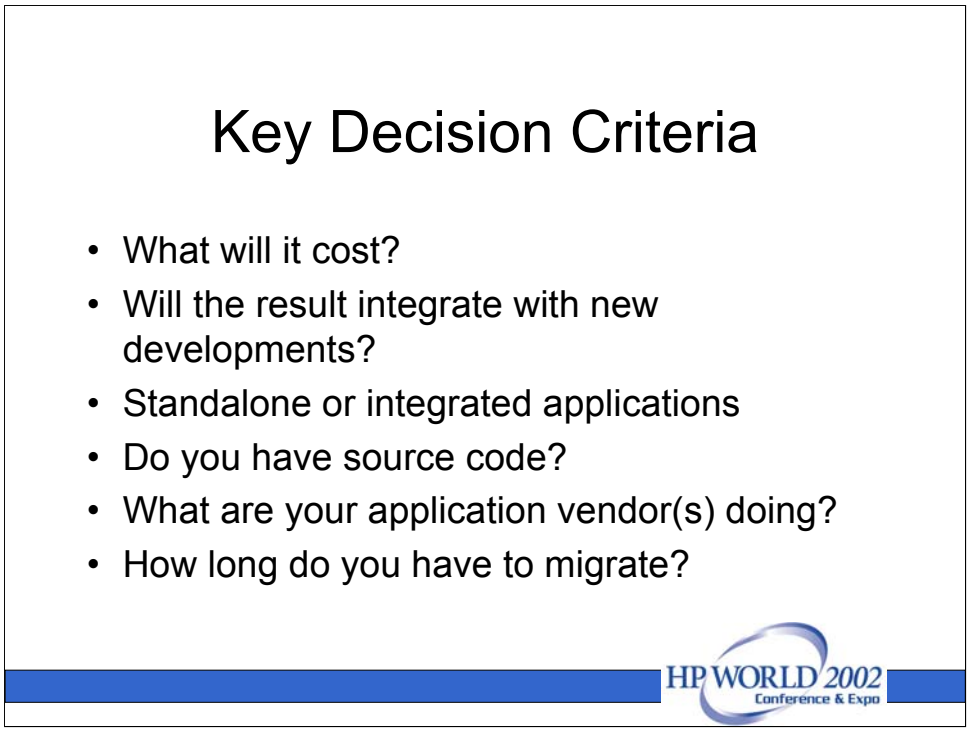
Re-engineer/Rewrite

Extract the business logic and rewrite using your chosen operating system, language, database, user interface. I do not intend to cover this in any depth as the variables are too great, and I do not count it as migration. However there are migration service companies out there that do have tool sets that they claim help them extract business logic from existing code, creating business objects, which they can then use to re-write an application using a variety of languages and databases.



Key Decision Criteria

Big Bang or Gentle Transition
Where do you want to end up?
Will you be managing the migration?
Maintaining the migrated solution
What in-house skills do you have?
What external skills are available?
What tools are available?



What will it cost?

Will the result integrate with new developments?

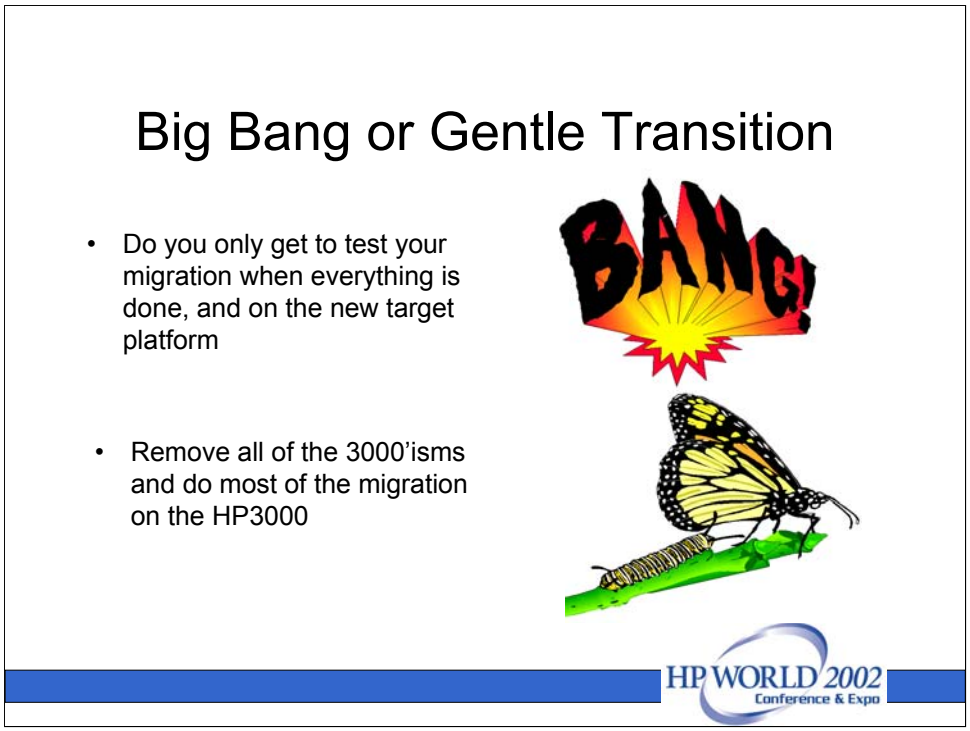
Standalone or integrated applications

Do you have source code?

What are your application vendor(s) doing?

How long do you have to migrate?

We will look at these criteria and try to find out where the Fossilization and Transformation approaches either facilitate or hinder.



Big Bang or Gentle Transition?

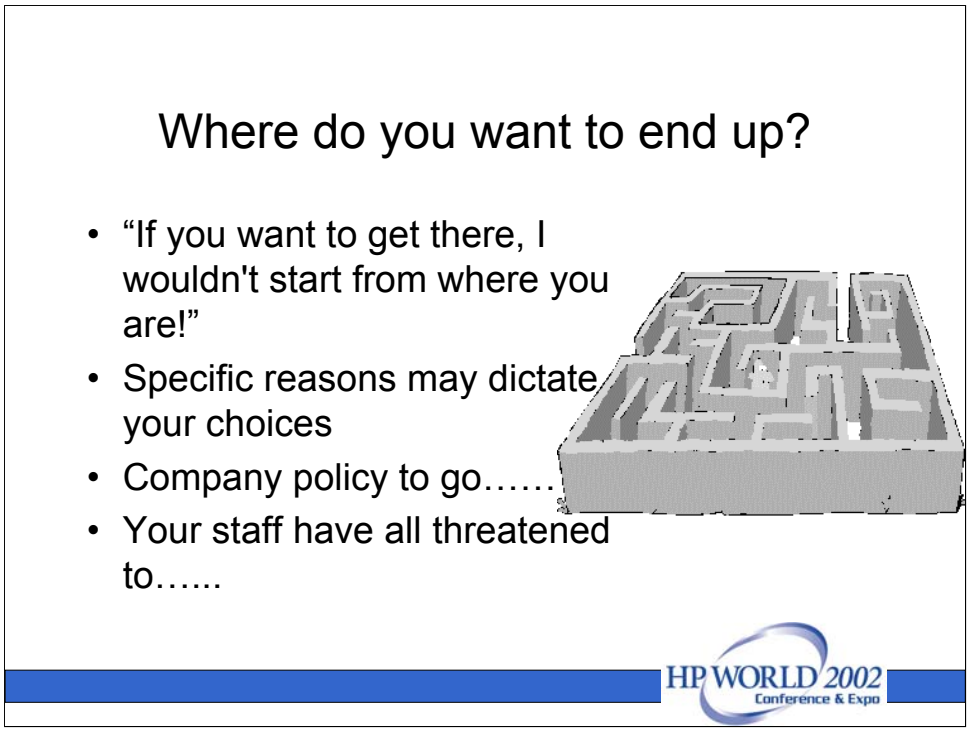
This could be a session in its own right…….

Do you only get to test your migration when everything is done, and on the new target platform, or do you remove all of the 3000'isms and do most of the migration on the HP3000. Both are practical and possible.

Big Bang is good in that it gives you ultimate flexibility in what solution you choose as it only has to work in the new environment. The downside is that you have an awful lot more to test at one time, and isolating where the problems are will be harder as more components will have all changed at once.

Gentle Transition is good in that much of your migration work can be tested and live long before you migrate. You may actually get some immediate pay back in things like a better user interface. The downside is that if you want to do this, it will restrict some of the choices about what tools and services you can use.

Gentle Transition can mean that application by application or even subsystem by subsystem, things like: moving to a non HP COBOL such as ACUCOBOL; replacing the user Interface; shadowing your data to a different DBMS on another platform and gradually re-writing your conventional 3and 4GL



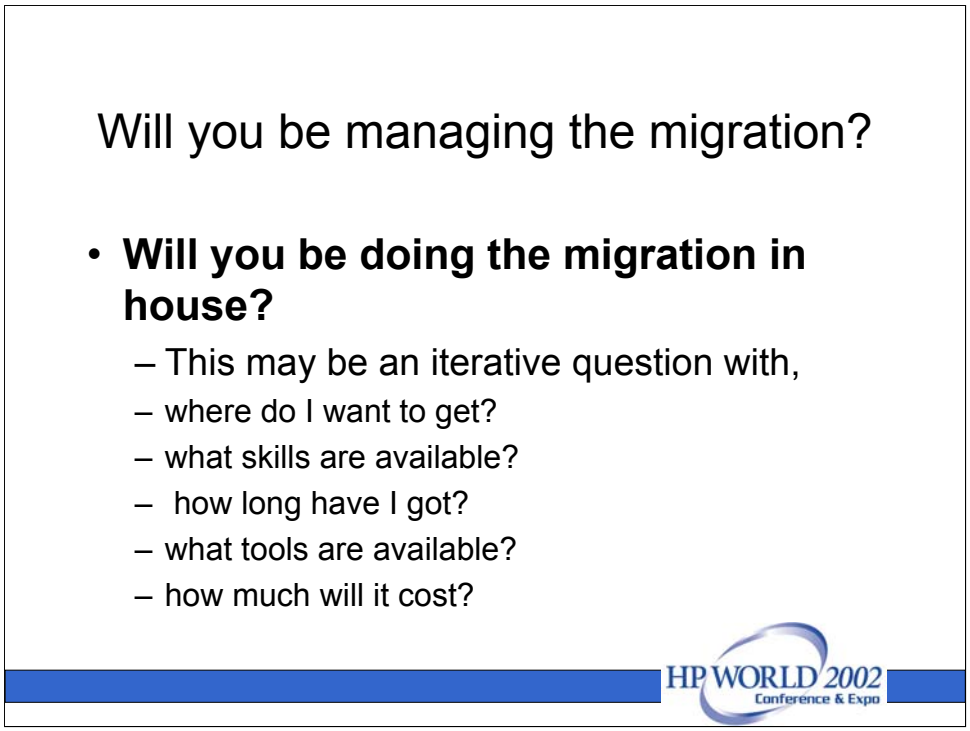
Where do you want to end up?

There is a wonderful old (Irish?) saying: "If you want to get there, I wouldn't start from where you are!"

Apart from the obvious answer "Somewhere different" (I know most of us would rather have stayed where we were L) you may have specific reasons that will dictate your choices. It may be company policy to go "NT", "HP-UX", "LINUX" etc, you may have made the decision that only …….. DBMS is acceptable, your staff have all threatened to quit unless its written in …….. and uses ……, or vice-versa have threatened to quit if it is written in …… and uses …….

Fossilization techniques certainly do not preclude some of these elements, the Emulation approaches do support a number of DBMS options, and operating systems, whilst the language choices are likely to be less flexible as you are buying into an environment that they have developed to achieve a specific outcome, the recreation of your HP3000 environment somewhere else. (go to cost and time questions)

Transformation offers a wider variety of choices in all these areas, all you have to do is find someone to do it for you (go to cost question) or find the tools to help you do it and the human resource necessary (go to what tools, what skills



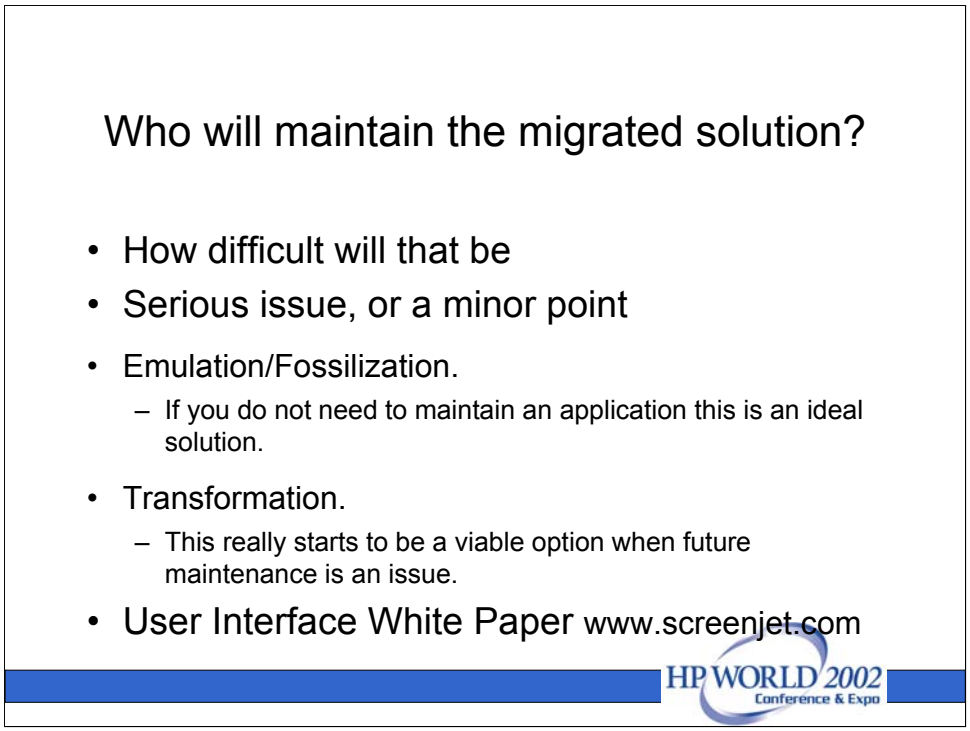
Will you be managing, doing the migration in house?

This is almost certainly an iterative question with, where do I want to get? what skills are available? how long have I got? what tools are available? how much will it cost?

Fossilization. At the time of writing (although this may have changed by September) the Fossilization route does not provide a mix'n'match set of components that you can purchase and use. You are basically contracting for a service where someone will deliver to you a working faithful a rendition of your old application and the necessary software components necessary to achieve that. Their reputation will depend on the word working, so don't expect to be able to get your hands too dirty.

Transformation. As this year (2002) has progressed we have started to see a number of migration tools emerge to assist with migration projects: Language converters (COBOL to COBOL), (COBOL to JAVA), (COBOL to C), (SPL to C); Database converters (IMAGE to ORACLE, SQL, DB2, ELOQUENCE); VPLUS converters (VPLUS to ACUCOBOL), (VPLUS to HTML), (VPLUS to VB). And to further support migration software that will allow the migrated code to retain IMAGE, VPLUS, and INTRINSIC calls.

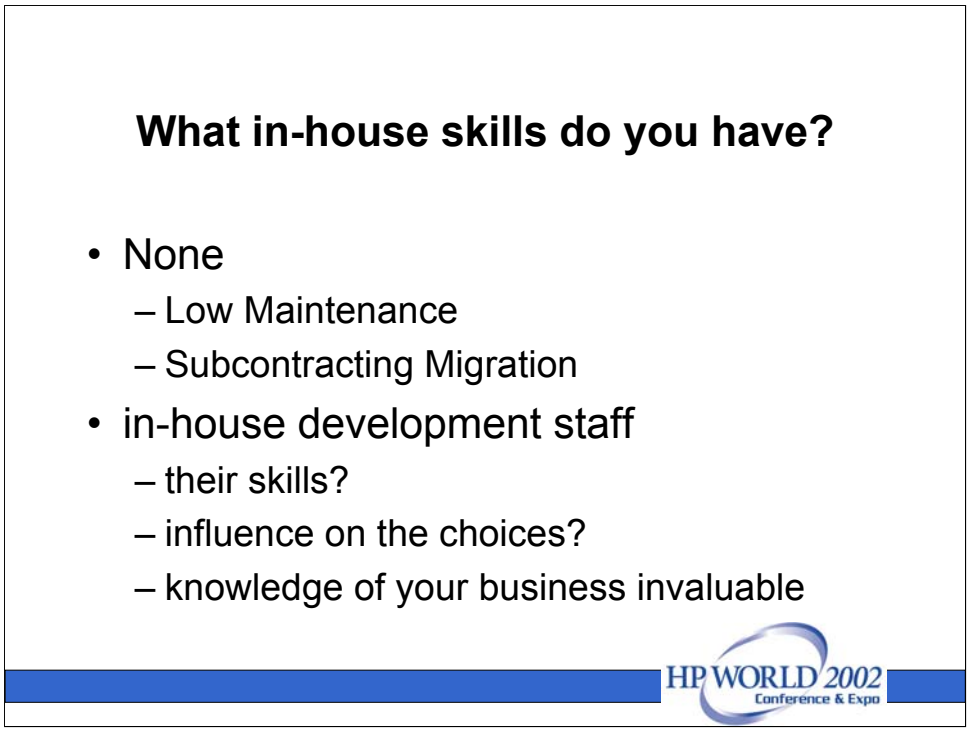
So when it comes to in-house migration Transformation offers a reasonably wide set of choices, but nowhere near as many as Re-Write, where the world is



Who will maintain the migrated solution

This may either be a serious issue for you or a minor point. If you are migrating an application that never requires maintenance, either because it is used purely for archival purposes, or its done what its done for the last decade without modification. Then how easy something is to maintain is not a major issue. However if you are making constant changes to your applications, how easy it is to make changes, and what skill sets will be necessary will be a critical issue. This may be further extended if after you have migrated an application you want to start incorporating features from your target platform, or new facilities as they come to market.

Emulation/Fossilization. If you do not need to maintain an application this is an ideal solution. Your application is running on a new platform, it looks the same, works the same, but depending on your choice of database you may have to invest in some skills acquisition in this area. If you need to make frequent changes to screens, program logic, databases then this is going to be one of the important things for you to investigate when looking at potential Emulation solutions. Just because something runs like it did before, and looks like it did before, it doesn't mean that it will be as easy to maintain as before. For example some Emulation solutions convert screens to UNIX character based screen driver files, which will make Formspec on the HP3000 look like a "state of the art" Development Environment. Integration with new facilities on your target platform may also be an issue. So now your on your nice new

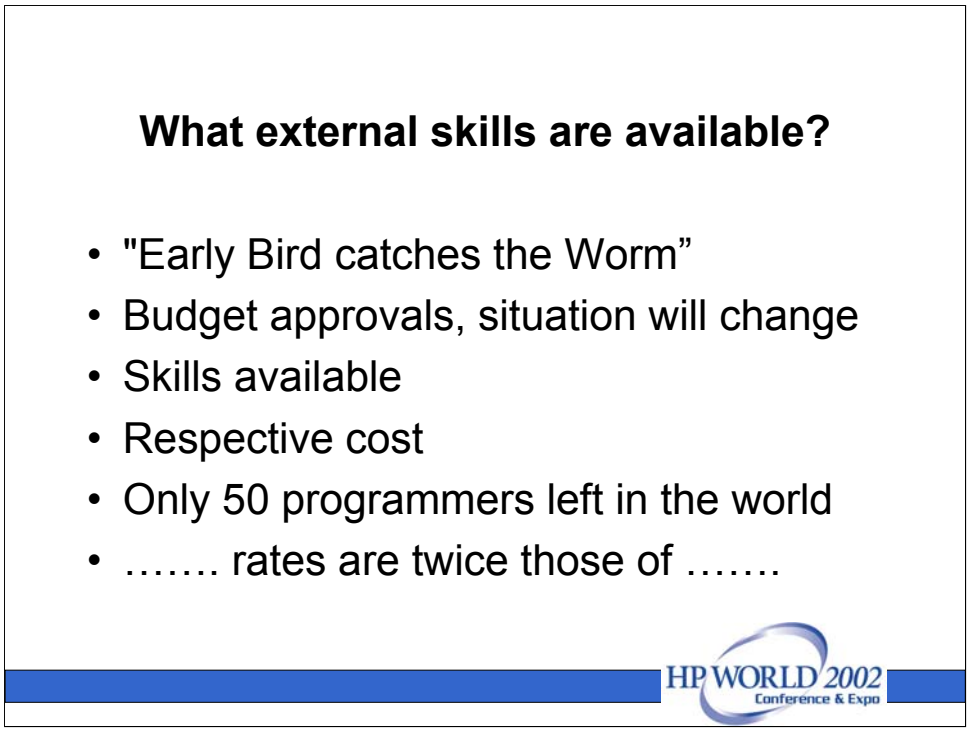


What in-house skills do you have

If the answer is none, then you are almost certainly in a low maintenance environment where any of the change issues are irrelevant. You will probably be subcontracting the task of migration, therefore the choice of what languages database etc are likely to be subordinate to, Who are you going to get to do it, What will they charge to do it, What will they charge to maintain/support it afterwards.

If you have in-house development staff, then their skills are likely to have some influence on the choices you make. Whilst pure HP3000 shops where skills sets are restricted to MPE with COBOL/IMAGE/VPLUS, TRANSACT, PowerHouse / Speedware, are less prevalent than in the past they are still common. One point I would make about your existing staff is that their knowledge of your business and how the current application(s) work will be invaluable in any migration project. Also do not dismiss their ability to learn new things, and new languages. When we used to do a lot of project development work for companies, we always used to reckon that any in-house staff allocated to the projects, even when they had to learn new languages, would be a fully productive member of the team within 3 months.

So if we are looking at migration solutions that buy into their existing Skills like COBOL or Powerhouse then you are likely to hit the ground running.



What external skills are available?

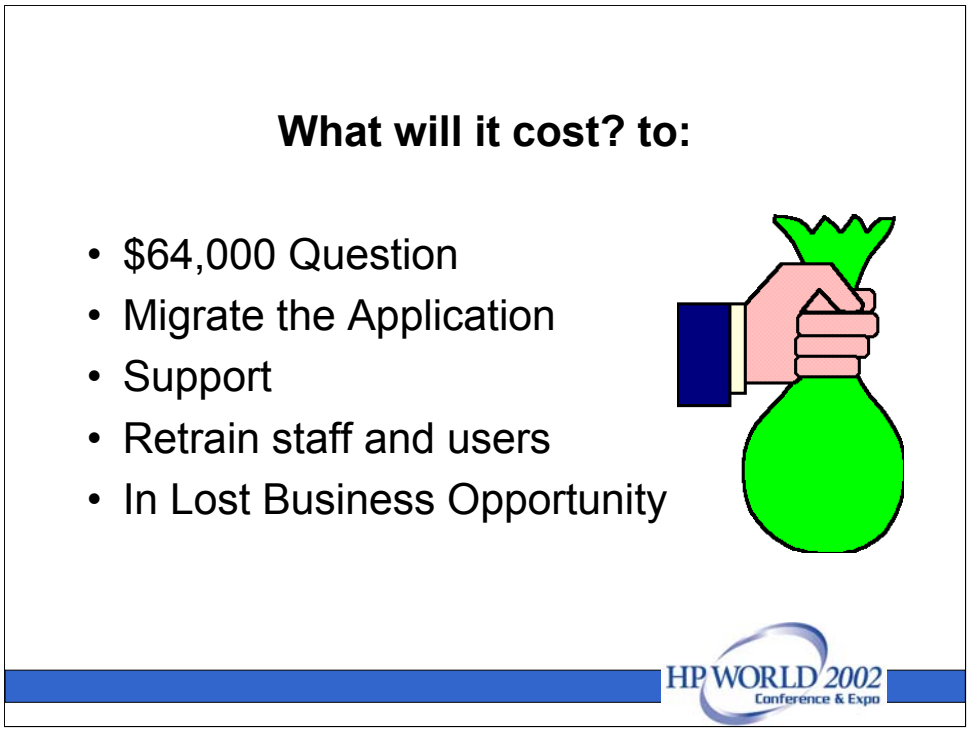
If you don't have the internal resources to undertake a migration project, then you are either going to be subcontracting the project, or hiring staff. In the case of the former you can now move on to the, When and How Much questions. At the time of writing there were more companies looking for migration projects and work than you could shake a stick at. As time moves on and budgets start getting approval expect that situation to change. The "Early Bird catches the Worm" if you actually want to catch the Worm however is a different question. If your looking at hiring staff either on a contract or permanent basis, then the skills available, what they might respectively cost, may well be contributing factors in some of the technical choices you need to make. Its not much good selecting a conversion strategy that turns COBOL programs into MUMPS only to find out there are only 50 programmers left in the world who know that language. Or perhaps COBOL into C++ only to find that C++ rates are twice those of COBOL.

What tools are available

It is my belief that tools will be the critical element in most migration projects, whether it be tools that migration service companies, have, develop or buy in that will help them undertake migration projects, as reliably and cost effectively as possible. This will apply equally to both Fossilization and Transformation services. Or tools that will make in-house migration projects more quantifiable, and the outcome more predictable.

A conversion tool that someone has already developed and used, is going to bring with it a certain amount of predictability about what it will and will not do. Also you will be buying into the experience that other people who have used that tool will have fed back into the product, so by the time that you migrate you will be some steps removed from the bleeding edge.

Since November 14th there has been a constant trickle of new tool announcements from companies looking to leverage their knowledge to make money. There is now a potential market that makes the considerable investment required in developing some of these tools worthwhile. This is going to work to your advantage as you are going to be able to buy tools for 10's of $1000's that will undertake tasks that previously would have cost 100's $1000's to have done before, or to do manually. This is probably one of the major differences between pre and post Nov 14th tools. Previously migrations were actually few and far between, were major projects, either brought about



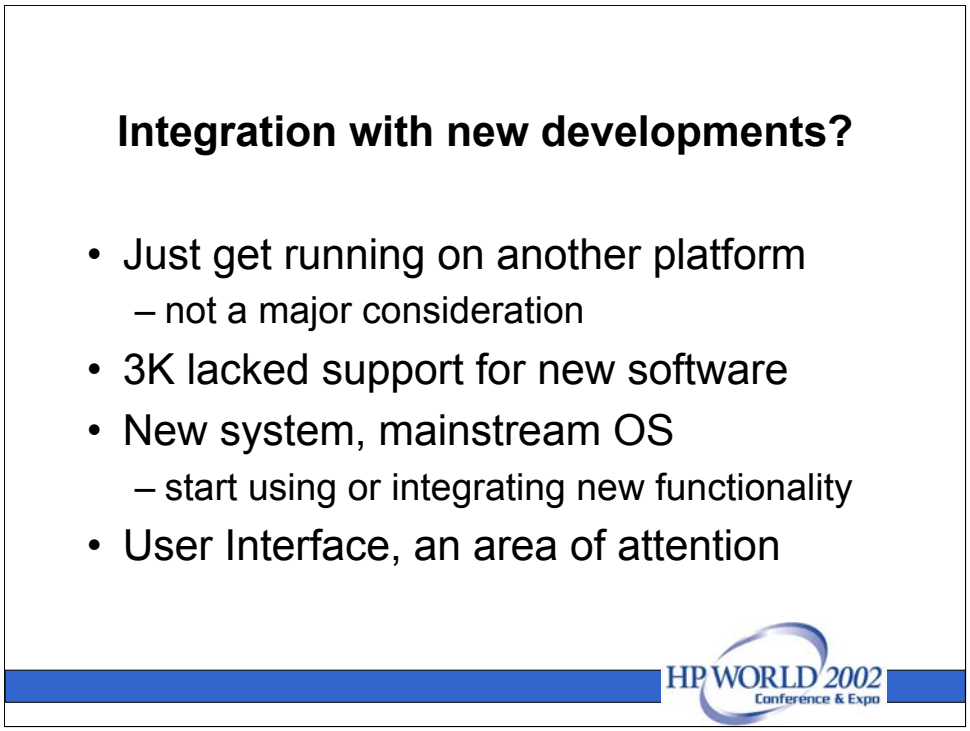
What will it cost to migrate

This of course is the $64,000 Question, or perhaps more L. There are really four major cost components to migration, ignoring hardware issues. 1) The actual cost of doing or having the migration work done 2) The Cost of training staff and users 3) The support costs once you have migrated 4) The lost business opportunity cost whilst you migrate.

Actual Cost

With the Emulation/Fossilization approach this should be fairly quantifiable, as you will be getting someone to quote for delivering a specific piece of work and software, to which you only have to add the cost of any additional DBMS, Utility software etc. Go and get your budget approval, and agree with your supplier payment terms and delivery schedule. About the only additional item you need to factor in is testing. So whilst the cost may be high, it may be largely fixed and concentrated into a short period.

With the Transformation approach costing is going to be a more complex exercise, unless you are subcontracting the whole task. To start quantifying the costs you are first going to have to work out where you want to get, what tools are there to help you achieve that goal, how much man effort is going to be required, over what time period will you be doing the work, will you be using existing internal resources or additional contract resources. In addition you



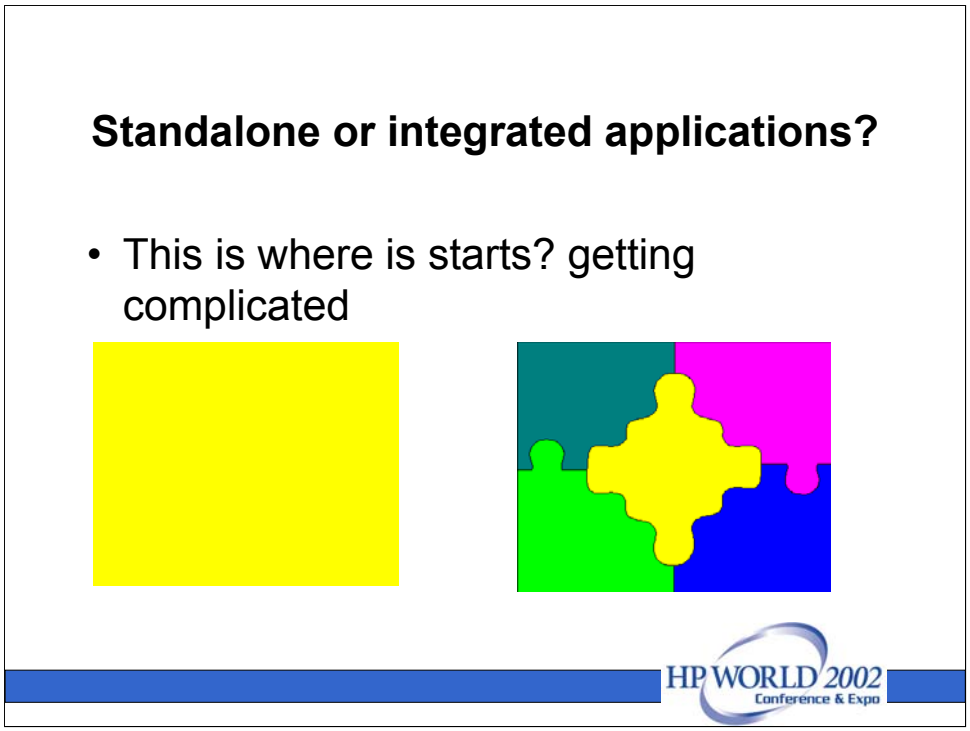
How well will the result integrate with new developments

There are two areas to consider here. If your main ambition was just to get what you had running before on another platform, then this need not be a major consideration.

The main downside to the HP3000 was its lack of support for new software as it was developed, and the fact that very few software developers ever ported things to it. (apart from the sterling efforts of a few individuals who did a tremendous amount of POSIX porting). In the main we had to wait years for either HP to write features into MPE/iX or for 3rd parties to write tools.

Now you have your application migrated on to a bright new shiny system with a mainstream operating system, where there is lots of software and new developments arriving all the time. Are you going to want to start using or integrating any of those things with your migrated application?

Fossilization. This is almost certainly a NO, you have your application running in an Emulate MPE shell that is doing its level best to do what MPE did. I believe that the chances that this shell will be enhanced to facilitate either the integration of new external services with your application, or vice-versa must be slim, just remember how difficult this was for HP. The area where this may show up the most, is likely to be in the converted VPLUS User Interface. Here you should be grateful that someone has managed to faithfully replicate the

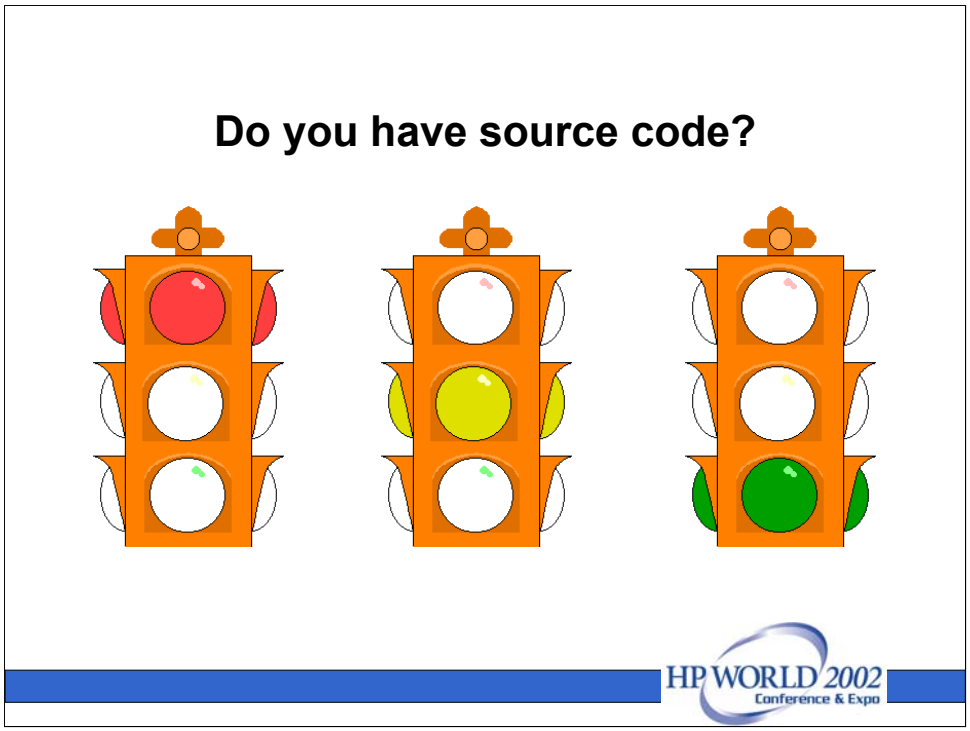


Standalone or integrated with other applications

This is where is starts? getting complicated. If you have a standalone application then your choices are virtually limitless. However in most environments there are a number of applications and the links between then can be quite closely coupled. If they were all written in house and in the same languages etc, then fine, or if the links between them are clearly defined and done via mechanisms that inherently don't really care about where the other application resides, then you will have more flexibility. But what happens if you have written a superb in house Order Processing system in One Language around a Manufacturing system written in another. Even if you can migrate both, you may well find that technically the best solution for each individually means that the result is no longer capable of the integration they had previously.

If all your applications are in house written there is probably very little difference between the Fossilization and Transformation routes in this respect. Perhaps the main difference will be in the Big Bang or Gentle Transition question. Where if you can undertake the bulk of your migration work on the HP3000, Transformation may well enable you to manage the migration of multiple applications as a serial rather than parallel process. Making the final leap to another operating system much less of a major hurdle.

If your applications weren't written in-house……



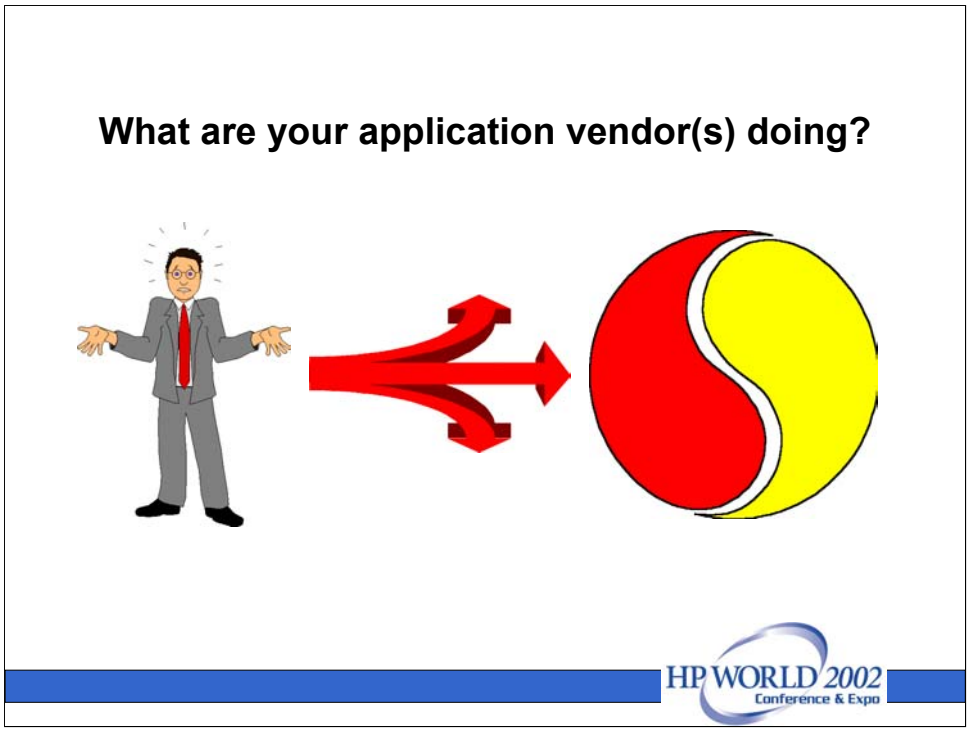
Do you have source code

No! Migration is not an option, Go to Jail, Do Not Pass Go, Do not collect $200. The choices I think you have are 1) Go find some new software, 2) Hope that the HP3000 Hardware emulation project comes to fruition 3) Have someone take your existing application and reverse engineer the logic, and rewrite it or you.

On this last option, much to my surprise and against my predictions, we had a customer in a similar situation with an old COBOL/VPLUS/POWERHOUSE application, for which there was very little documentation, and a lot of the source code had been lost when the company shed all its IT Staff. They had an Indian company sit down, document it, trawl through the databases etc. This company then went away and rewrote the whole application (Well the 80% that was needed), in JAVA, on Linux using an Informix database. It ended costing more than the estimate, and took twice the planned time, mainly due to the constant testing/re-coding iterations. But they got there, the customer is happy, and the application is now happily integrated with applications on the HP3000 using FTP, JDBC, and SQLLink.

You do have the Source Code! But do you have the rights to the Source Code?

This is certainly too big a topic to cover now, however just having the source code doesn't mean that you have the rights to use it to migrate the application



What are your application vendor(s) doing

If you have major applications from vendors what are they doing about providing a migration path? I will not go into the issue if staying with that vendor is the right or wrong choice for your company, but if you are staying what they are doing may well affect your whole strategy (and interestingly at this point in time what you plan on doing may still affect their strategy).

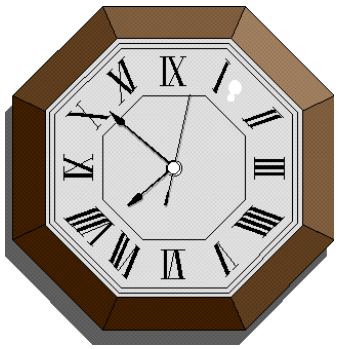
If you have integrated applications from multiple vendors, and one has a migration path to HP-UX and another to Linux and you quite fancied going NT, there is obviously going to be a parting of ways at some point. Even if one of them did offer a migration path that fitted your strategy, and the other didn't, the outcome may well be that you will go with neither and select completely new software.

However my main point about what are your vendors doing, what not primarily about whether you stay with them, but rather technically what route are they taking.

If as in many companies you have done extensive customisation around a vendor's application, using the same language and user interface that that application was written in. Then looking at what tools, and what development environment they are using (or still at this stage planning to use) may well mean that you can buy into their research and the expertise they gain, when

How long do you have to migrate?

- This is something that only your organisation can decide, based on your business requirements, and your views on the risks involved with the different migration and homesteading strategies.



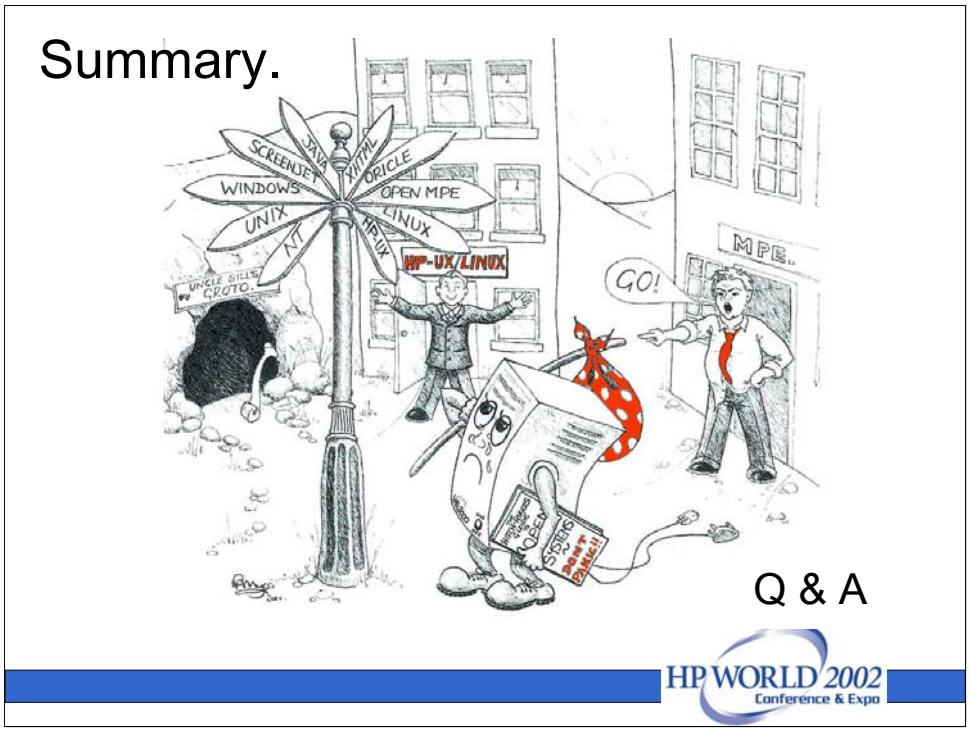
HP WORLD 2002 Conference & Expo

How long do you have to migrate

This is something that only your organisation can decide, based on your business requirements, and your views on the risks involved with the different migration and homesteading strategies.

Fossilization. If you have to go soon then Emulation may be your best strategy. All you have to do is agree cost and timescale with your chosen supplier. Also if for some changing business reason in the future you suddenly have to make a move, this may be a good back up strategy. Provided that is you are not halfway through another migration strategy.

Transformation. Time scales for this approach are rather like a piece of string. We have already this year seen several very rapid conversions of COBOL, VPLUS, IMAGE applications using conversion tools. However it would be fairest to say that this approach is particularly suited to either migrating a standalone application, on in more complex environments where you have come to the conclusion that you can take your time over the migration, particularly if you can buy into the further advantages of Gentle Transition rather than Big Bang.



Summary.

Nobody can make the decision for you which of the Fossilization, Transformation or Re-Write options are best for you. And almost certainly just because company X has chosen a particular route does not mean that it will right for you. There are a number of companies who are developing a depth of experience in evaluating and advising on migration strategies. They may well be able to assist you in evaluating your choices, and advise you on what works or hasn't worked. My personal advice would be don't take any advice about what the best migration choice for you is, from anyone who can only offer you one solution.