OpenVMS to Linux and UNIX Migration

Performing a Managed Migration

In companies across the globe, IS managers are looking toward the future with varying degrees of apprehension. In many cases their comfortable world has been disrupted by a mandate to move from their familiar operating system into is the brave new world of UNIX. For most OpenVMS managers the road from OpenVMS to UNIX does not look smooth. A gap exists, the void between the existing OpenVMS business solution and the new land of open systems. The managers' OpenVMS knowledge base seems useless for addressing the problems they could face in the coming months. Familiar commands no longer work. Text editors seem to have been designed to defeat the editing process rather than enable it. System management tools are nowhere to be found. Even HELP fails to provide information. It's no wonder IS managers don't sleep soundly.

But these fears do not have to be fully realized. You can implement specific strategies to help manage the transition from OpenVMS to UNIX. Several companies market tools and services that make the migration of both application code and end users cost effective and timely. End user migration tools can also aid in situations where OpenVMS will coexist with UNIX, enabling a consistent user interface (see "Compiling Your Migration Toolset," page 28).

Starting Out Right

Two primary reasons migration projects come in over budget or behind schedule are incorrect planning and insufficient risk management. Incorrect planning stems from using planning skills based in the software development paradigm instead of a paradigm developed specifically for migration projects. Insufficient risk management results from not incorporating the risk components directly into the migration methodology.

> *For better or worse, a large number of Open VMS system managers have been issued a corporate mandate to migrate computer operations to UNIX. Here's how to embark on a program that will efficiently migrate your code, personnel and overall IS resources to a UNIX operating environment.*

5 Stages of UNIX Migration

ASSESSMENT. *By understanding and detailing the project size and scope, you reduce the risk of the project's going over budget.*

PLANNING. *The planning stage provides a technical architecture for software identified in the assessment stage.*

PORTING. *During the porting stage, your UNIX system is created. The major effort of this stage is to obtain a working system that can be demonstrated.*

VALIDATION. *The validation stage can bring the longest term value. The final deliverable is your OpenVMS solution on open systems with your users working with it.*

PRODUCTIZATION. *The final stage is productization. The productization stage converts an OpenVMS solution running on UNIX into a true open systems solution.*

Planning

IS groups moving from OpenVMS into the open systems environment require new skill sets. While the goals of most activities are the same as those for any operating system, the specific actions you take are different. Be prepared for changes in the areas of system and network management, accessing the email system and using the editor, for example.

Migration of critical corporate systems requires coordinating many diverse activities, including hardware resources, user documentation, training of the IS staff, and training of the trainers. You also need to address other activities such as beta testing and product release, which may not have been done for years.

Don't forget to plan for intangible issues as well. One striking effect of a migration project is the cultural change the IS staff experiences. OpenVMS and UNIX view the world in significantly different ways, and this can affect morale and productivity.

One item sometimes overlooked in the scramble to open systems is the impact on customers. Mandating a change in operating systems often has the same effect in an organization as mandating a change of religion would have in a community. You must develop a migration plan that is acceptable at all levels of the organization. In addition, you must consider both external and internal customers. Your customers are accustomed to using your software in an OpenVMS environment. Their first question will be, "Are you abandoning us?"

To make the transition as smooth as possible, clearly explain reasons that support the change. Nothing will cause more problems in a migration than an organization that does not fully support the effort. The same people who *understand the applications and the process today are the people who can add the most value in porting the applications and migrating the users.*

Risk Management

To effectively embark on a migration project, you must develop a risk management plan. By identifying the problems and outlining strategies to deal with them, you can find solutions to all potential problems. If a demonstrated solution is not available, you may have to develop a worst case scenario prototype to verify the solution. There are essentially five stages to a UNIX migration. These stages correspond to five components of project risk management and should allow you to migrate effectively while maintaining and potentially enhancing your existing software and personnel investment. Key to migration is safeguarding your investment in those business critical applications encompassing endless person years of corporate knowledge and training while dealing with the proprietary parts of the applications. A migration project that addresses the five risk management components upfront will likely be delivered within budget and on schedule (see box, "5 Components of Migration Risk Management"). The methodology is based on five stages, which can run sequentially or in parallel. The purpose and goals of each stage remain the same on all projects; only the size and details change from project to project. The five migration stages are each linked to a component of migration risk (see box, "5 Stages of UNIX Migration"). Solid management and control of each stage can make migration a breeze.

Assessment: Reducing Cost Risk

A complete project assessment is the first stage of any serious migration project. By understanding and detailing the project size and scope, you significantly reduce the risk of the project's going over budget. In the assessment stage, you must develop a detailed understanding of:

1. The existing corporate and computer environment.

2. How software is developed.

3. The technical environment in which the applications exist.

4. The target environment.

The migration process itself provides opportunities for the company to perform a fairly detailed situational analysis and should result in a clarification of the organizations IS goals and direction. You can perform assessments in two steps, a quantitative analysis followed by a detailed risk analysis. No matter what format your assessment takes, at its completion it should account for every work item for the migration, and no unknowns can remain that could drive up the final project price.

Planning: Mitigating Schedule Risk

While the activities of a migration project are similar to those of a development project, the slant is different. First, you must provide a technical architecture for all software identified in the assessment stage. For any items without demonstrated solutions, you must develop and verify small prototypes.

The second step is to outline specific activities to address the risks identified during the assessment stage. Address these risk items via avoidance, control, assumption or transfer.

Risk avoidance, used at the beginning of a project, removes the cause of the risk. Avoidance may involve de-scoping the project or changing some significant factor involved.

Risk control, performed during the migration project, involves intensive management intervention. It may include activities such as parallel migration subprojects or frequent management reviews.,

Risk assumption involves going ahead as planned, accepting that there is the potential of not meeting the schedule or going over budget.

Risk transfer involves removing the source of the risk from the migration project. This could mean delaying the high risk elements of the project, outsourcing parts of the, project or hiring outside consultants. Although outside consultants may appear to add unnecessary cost to the project, their objectivity and experience joined with the knowledge of the in-house staff can result in substantial savings. The result of the planning stage should be a schedule with achievable, tangible and measurable milestones. The first major milestones will be configuration management and revision control and a reliable build of the application on the target environment. A rapidly moving migration project depends on the ability to have incremental builds on a daily basis. Remember to plan for testing. Full system testing begins the day a successful build is available.

Porting: Managing Technical Risk

In the porting stage of the project, your UNIX system is created. The major effort of this stage is to obtain a working system that can be demonstrated. At this stage, you implement the solutions identified during the planning stage. The UNIX based build becomes available; the emulation tools for DCL, FMS, system services, and so on, are integrated; and specific reengineering tasks are performed. Also at this stage, you must carefully watch configuration management and version control and the build mechanisms, because they can cause the project to fall behind schedule and be over budget. This is a terrific time to train people on UNIX, because they can immediately get hands-on experience with an application they are already intimately familiar with. The new UNIX

Migration in the real World

Sector 7 was recently involved with a migration process that used the stages outlined in the accompanying article. The client had a legacy system involving some assembler, 2.5 million lines of VAX FORTRAN in 10,000 sources files, and 12,000 lines of DCL code. The application included more than 1,200 individual functions users could access. These functions were made available using about 170 images. The entire application was designed around the strengths of the OpenVMS VAX architecture — for example, the use of the OpenVMS VAX parameter passing mechanism. Certain FORTRAN routines were designed to receive differing types of parameters. For example, you could pass to the same formal parameter an integer, a float or a string. For those not familiar with FORTRAN77, this is probably considered illegal in conservative parts of the U.S. The application accomplished this process by having the called subroutine receive a byte structure representing the OpenVMS parameter passing structure and then decoding the fields. The application was built on RMS ISAM files accessed through the system services, and they needed to deliver their application in Oracle.

The client made two attempts to migrate the application. The first attempt used a software translator to convert the application into C. The target system included hardware with a different byte ordering than the VAX and required all data to be even byte aligned. Needless to say, this attempt was not successful. After 10 months of the project the CEO went away for 4 days, and when he returned the project was 4 months behind schedule. At this point the company did a major reset on the project and spent weeks evaluating their options.

The result was a changeover of the majority of the project team and the suppliers of the software tools, and a completely new project plan. The new plan followed the methodology described in the accompanying article, and they delivered to the customer a reliable and fully tested system within 6 months.

While this may sound like a nightmare project, from my experience it is typical of most organizations attempting migrations. The wide availability of OpenVMS emulation software and the lack of experience combines to result in projects like this. Emulation tools significantly reduce the amount of rewriting necessary, but they can never replace good project management skills, experience and a solid plan.

staff can build credibility and become integrated into the corporate culture. The OpenVMS

staff members look to the UNIX staff for information and direction, while the UNIX staff look to the OpenVMS staff for application knowledge.

Validation: Controlling Operational Risk

The validation stage can bring the most long term value. It should begin when the corporation has committed to the migration project. The final deliverable of this stage is your OpenVMS solution on open systems with your users working with it. The major advantage of a migration project — the fact that the system is already working — is also potentially its largest problem. A working system provides practical opportunities. Specifically, a complete test suite can be available before the first line of code is migrated to the new system. The flip side is that the use of high level languages and reliable emulation software hides some of the dangers of a migration.

Components of Migration Risk Management
1. Cost. Will it be within budget?
2. Schedule. Will it be done on time?
3. Technical. Can it even be done?
4. Operational. Will it work?
5. Support. Can it be maintained?

There are a number of places errors can be introduced during migration: 1. Embedded application errors. 2. The compiler. 3. Hardware. 4. Fundamental differences between OpenVMS and UNIX. 5. Reengineered code. 6. Emulation software. However, when failures occur in migrated software, they fail bigtime, and there are no new logic errors. The errors are almost always obvious and are quickly found and fixed. You should employ automated testing with an emphasis on functional and regression testing. The final test set can act as your project acceptance criteria and provide a return on investment long after the migration project.

Productization: Minimizing Support Risk

The final stage is productization. Although it can be difficult to decide when

productization is complete, its purpose is clear. The productization stage converts an OpenVMS solution running on UNIX into a true open systems solution. This represents activities that range from dealing with performance issues to adding a complete client/server GUI.

MIGRATION CAN BE THE solution for organizations looking for a low risk introduction to open systems with a high payback. It is the type of project that is seemingly straightforward but actually quite complex. However, if you exercise proper care, migration projects can be delivered on time and within budget, and they can even be fun. If used appropriately by senior management, they can lead to a rejuvenated IS organization making better use of financial resources

The Experts

Since 1985 Sector 7 has been providing solutions for companies needing UNIX solutions that maximize their software investment in OpenVMS. We have encountered all the pains and pitfalls involved in moving from a proprietary environment to UNIX. We understand the migration process, we have benchmarked it, and we have improved it.

The tool-sets developed by Sector 7 provide the solution for a number of obvious technical problems when moving from OpenVMS to UNIX. These problems, while significant, are only part of a large migration project. Before embarking on a significant migration project an organization must be aware of potentially significant challenges in project management and project implementation. Unfortunately, lack of migration experience causes these challenges to only become apparent in the middle of a migration project. When expected and planned for, these challenges become opportunities for organizations to learn, grow, and improve.

The Sector 7 Professional Services group is a world-wide organization designed to complement your migration effort. Our broad range of experience allows us to be effective in all aspects of a migration project. From the business case to the most detailed technical problem Sector 7's expertise can help guide the process and avoid risk.

A large number of organizations choose to migrate their applications using their own resources. This is most often the case when software is critical to the organization and the port will incorporate significant future enhancements. In these situations, the organization is involved in all levels of the work with Sector 7 providing the support needed to ensure a successful migration.

Alternatively, many organizations have several concurrent projects with limited staff resources. A migration will bridge the gaps until new applications are brought on line. These organizations will use Sec-tor 7 to perform the complete migration project and provide a validated version of their application in a UNIX environment.

Migration Consultation

The Migration Consultation is a two day program giving organizations an under-standing standing of migration and determining if it is an appropriate solution for them. The first day consists of four seminars explaining the migration process and the relevant issues. The second day involves focused discussions designed to understand the key particulars of your application and your needs.

Application Assessment

Sector 7 has a standardized Migration Assessment Procedure that reviews over two hundred different technical items in five groupings. Included with the assessment is a formal analysis of the Technical, Cost, Schedule, Operational, and Support Risk of the migration project. Our assessment concludes with the delivery of a detailed report. This provides, project analysis, technical strategies, a breakdown of the required work, and the expected risk management efforts.

Migration Planning

Migration planning addresses some obvious and some non-obvious technical de-tails. Included are the Human, Software, and Hardware resources. Deliverables include the normal project schedule, resource requirements, and most important for in-house migrations, the beginning of involvement and ownership for the existing IT staff.

Sector 7 has performed over 1000 application migrations world-wide. Each one has been a learning experience, not only for Sector 7, but for the client as well. You can use our experience to create plans and schedules based upon achievable, tangible, and measurable milestones.

Porting

During the Porting Stage, the code is moved from the OpenVMS platform to the target platform. The issues revolve around technical problems and their solutions. At this point minimum re-engineering of specific applications can be performed and new database technology introduced. When completed, all technical problems between the original and migrated applications will have been uncovered and solved. The focus here is following, meeting, and beating the project schedule.

Application Validation

Application Validation is critical and, if used effectively, provides the most significant reduction in the cost of delivering a project. In a migration, the final deliver-able exists before the project has begun. When used as a leverage, test and validation can be exploited through the entire project. Using our expertise in the five areas where a migrated application can fail, Sec-tor 7 ensures reasonable and appropriate AUTOMATED test plans are developed.

When the code is ported to the targeted UNIX platform, the final phase of the migration is at hand - putting the code into the hands of the end users. In most cases the application programs will function exactly as they did when running under OpenVMS. The only changes that the users will notice will be in the basic tools that are operating system dependent.

For OpenVMS users moving to UNIX, these tools will be completely foreign. To make the last phase of the migration succeed there are two options. Retraining the entire user base so they can use the native UNIX tools or providing OpenVMS productivity tools to duplicate the OpenVMS environment on UNIX.

OpenVMS productivity software allows the users to continue to be effective and pre-serves the investment already made in OpenVMS. In cases of heterogeneous environments these tools reduce frustration by providing a uniform interface to a variety of platforms.

Sector 7 provide the most complete set of migration and interoperability tools and services available today.