Facilitating Simultaneous Remote Data Center Migrations of Multiple Interrelated IT Systems

PML 2009

Amar Kumar Behera

Correspondence:
Project Manager
MphasiS an EDS company, ITO PPMG, Tower IV, Level VI
Cybercity, Magarpatta, Pune - 411013

Tel: + 91 20 4014 4575
Fax: + 91 20 6642 5003;
E-mail: amar.behera@eds.com

ABSTRACT

Datacenter migrations for a large corporate entity with a large number of IT systems (typically > 20), where a significant number of these inter-related systems are to be migrated within a stipulated contract duration, calls for project and program management in a complex framework. Issues related to multiple stakeholders spread across several countries, including from different corporate entities responsible for hosting, application development, application maintenance and outgoing hosting vendor present several difficulties. Besides, differing IT standards, movement to a new virtualized environment, compatibility of upgraded system components, reverse engineering of changes and scripting done in the lifetime of the system, managing change windows across systems, and communications management in the face of differing time zones and cultural and language barriers further complicate the situation. This paper discusses these issues with a real life case study, providing insights into making such a complex activity successful through effective application of PMI's project management framework.

Keywords: data center, migrations, stakeholder management, delivery, project management

INTRODUCTION

In a fast moving dynamic corporate world, faced with ever increasing capacity demands, power and space shortages, and introduction of new technologies with ever shorter product lifecycles presenting substantial issues of scalability, coupled with increased outsourcing and best shoring, and new networking paradigms, remote data center migrations have become a norm rather than an exception for many corporate entities. In this scenario, enabling these migrations within limitations of budget, time and space, presents several key challenges.

Such data center migrations may often be the responsibility of a hosting vendor, calling for a hand in hand co-ordination with the owner firm. Given the size of the owner firm's IT architecture, this may often mean several individual IT systems, each with its specific application, which may often talk to each other. This presents one layer of complexity.

A second layer of complexity arises in situations where the owner firm switches contracts between hosting vendors. This presents several facets to the migration task, such as dealing with the changes that the system might have undergone during the period the system was managed by the outgoing vendor. These changes may often involve custom built scripts which are proprietary and the outgoing vendor is unwilling to share. The challenge for the new vendor is then to build these in-house, often calling for reverse engineering, keeping in mind time and cost constraints. Again, lack of accurate and up-to-date documentation, at times, even complete lack of it, and unwillingness to share knowledge, could be a severe constraining factor in the progress of the migration project.

A third layer of complexity is often added by the presence of several stakeholders and multiple third party vendors and business units. For instance, there could one or more application vendors, individual vendors managing separate system components, hardware suppliers, software vendors, multiple departments within the owner firm having stake in the migration, project managers with each individual vendor, and the associated technical teams. Again, the interests of the user community while limiting outage time and maintaining user satisfaction is a key challenge. Managing communication amongst several stakeholders, while completing the project deliverables successfully could be a major cause for concern.

In addition, it is worthwhile to mention that all of this must be done while following organizational processes, with a clear focus on time, detail and budget. While the contract may impose certain restrictions on time, this should be realistic keeping the above constraints in mind. The migration process typically calls for a very detailed implementation plan, over and above any high level plans that may exist. The budget should take into account the detailed plans, and also account for any contingencies, disasters and rollbacks that may be called forth for. Updates to the project schedule on a regular basis, as and when needed, with effective communication to stakeholders is a must.

Keeping in mind the big picture painted in the above paragraphs, the remainder of this paper seeks to bring forth these challenges with the help of a case study, and generalize some key learnings and tools that may be helpful in

managing such projects better, and with an eye on higher success rates of remote migration projects.

CASE STUDY

The present case study discusses the IT infrastructure management of a large global conglomerate, who we shall henceforth refer to as 'Firm A'. Firm A's IT operations cover individual business needs such as e-commerce, logistics, operations, yield, infrastructure, etc. and consist of nearly 150 individual systems under the above heads. Firm A also maintains a dual vendor strategy across its entire business portfolio, including information systems. Until lately, a large global IT service delivery firm (henceforth called 'Firm B') was responsible for nearly 80 % of the above systems, while a medium sized local IT firm (henceforth called 'Firm C') was responsible for the remaining systems.

With the expiry of the contract duration with Firm C, Firm A took the decision to shift operations from Firm C to another large global IT service delivery firm, Firm D. This was with a view to account for enhanced capacity, new application releases demanding upgraded hardware and software components and rope in the added business capabilities of a global IT firm for enhanced performance, reduced down times, and efficient service delivery catering to a global user community. This shifting of operations of nine individual IT systems was conceived as a data center migration program with nine separate projects, with a time bound of 10 months. Each individual system again has individual third party vendors, providing application development, application maintenance, component development and management, hardware etc. With this general background, we next discuss the system architectures, components, requirements for migration, key challenges and how corporate best practices help in managing these projects successfully. To simplify matters, we will look at a single system comprising a single project, and see how it is affected in the presence of other systems, some of which are part of this migration program and others which continue to be hosted with Firm B.

System Architecture

We consider the system, which we shall henceforth call Sys A, which is a master customer data system for Firm A's transport line. This system is used to store, maintain and distribute customer data. It essentially consists of a web application accessed by the users using an internet browser. User authentication is done by SPNEGO (Simple and Protected GSSAPI Negotiation Mechanism) by looking up the Active Directory where user information is stored. Sys A provides message feeds to a host of external applications as well as receives message feeds from a few external applications. The message communication is achieved by using Message Queuing (MQ) and web services. There is also request/reply via MQ or web services using a message broker to process messages hosted by Firm E. The system is hosted on four environments, viz., development, test, pre production and production. Sys A has a contingency system which becomes active in case of an outage of the application in production, and pre-production. The total number of users expected is nearly 5000 with a peak concurrent users expected to be 200. The users form a global audience.

Sys A is hosted on a Solaris 9 UNIX OS with several components such as BEA WebLogic Server, IBM WebSphere MQ, Oracle Database Server, Hibernate, Struts, Log4j, Java and Castanet. The systems is characterized by a three tier structure as depicted in Figure 1. The presentation tier is based on Java Server Pages (JSP). Struts is used as the controlling component between the presentation and the business logic. The business logic is implemented using simple Java classes (i.e. no EJB's) in a service oriented structure. Only services in the business logic tier accesses the integration tier (which is based on Hibernate).

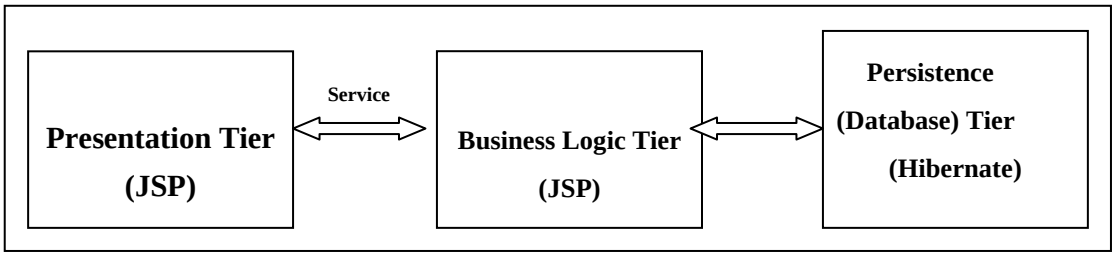


Figure 1: Multilayered architecture of System A

Migration requirements and dependencies

As many as 20 different systems connect to Sys A, of which three have dependencies with Sys A, as they are part of this migration program. We shall name these systems as Sys B, Sys C, and Sys D. Sys B only receives data feeds from Sys A, and is a smaller system serving primarily as a data warehouse. As such, it is a meaningful effort to migrate Sys B over to Firm D, prior to migrating Sys A. Sys C connects directly to Sys A using a request reply protocol, requiring two way interconnectivity. This in turn, calls for, steps in the migration plan, which take care of the message queues corresponding to Sys C. Sys D, on the other hand, shares virtual servers with Sys A and requires coordination in carrying out the server build and installation of software components. To properly understand and handle these and other dependencies, a new matrix approach was conceived where stakeholders, approach and risks are laid out against the dependencies (we refer to this as the DARS matrix henceforth).

Table 1: Stakeholder-Approach-Risk Matrix for handling dependencies

Dependency	Stakeholders	Approach	Risks
Message feeds from Sys A -> Sys B	Sys A owner (Firm D); Sys B owner (Firm D); Firm A PM*; Firm D PM (Sys A); Firm D PM (Sys B); Firm C Middleware Team; Firm D Middleware Team; Firm C SDE*;	Migrate Sys B ahead of Sys A; Inform Sys B users of outage; Ensure initial load of messages from Sys A ahead of migration; Stop queues at Firm C and start at Firm D during migration	Corruption in initial load due to message index errors; Message Queue stop/start leads to message loss
Request reply feeds between Sys C and Sys A	Sys A owner (Firm D); Sys C owner (Firm D); Firm A PM; Firm D PM (Sys A); Firm D PM (Sys C); Firm C Middleware Team; Firm D Middleware Firm E Message Broker Team;	Stop and start queues at right times during migrations of both Sys A and Sys C; Inform users of both Sys A and Sys C of outage during migrations; Introduce steps in migration plans to stop and start feeds at Firm E message broker	Message Queue stop/start leads to message loss; Communications management among 4 firms during checkpoints prior to and during migration;
Shared virtual servers between Sys D and Sys A	Sys A owner (Firm D); Sys D owner (Firm D); Firm A PM; Firm D PM (Sys A); Firm D PM (Sys D);	Ensure compatibility of installed components between systems; Make schedule updates in both Sys A and Sys D projects to ensure migration dates are such that both systems are available to users	Version and patch level incompatibility leading to system errors; Schedule mismatch leading to system outage, user dissatisfaction; Coordination between the two project teams;

	Firm D Technical Team; Firm D Facilities Management Team;

** PM – Project Manager*
** SDE – Service Delivery Executive*

Apart from the above dependencies, shown in the DARS matrix, associated with the task of migrating multiple systems which are interrelated, there are a number of other dependencies that are associated with the migration of each environment. These dependencies also need to be considered to develop the migration implementation solution, project management plan and project schedule. Some of these dependencies can be found in Table 2.

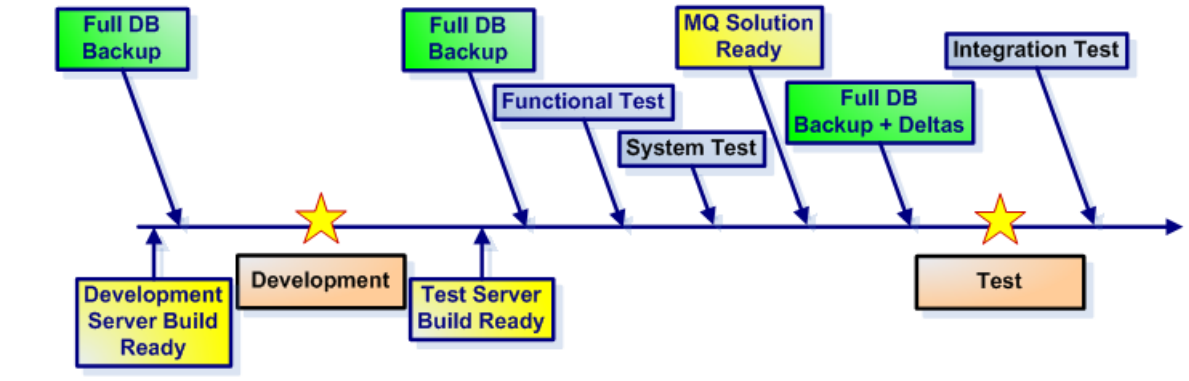
Table 2: Environment specific dependencies

Environment	Dependencies
Development	Before testing, a full database backup of Firm C's development database is required
Test	• Development testing is complete • Before testing a full database backup of Firm C's Test database is required • MQ migration solution in place & MQ Cluster scripts available from Firm E • Full database backup of Firm C's Test database is available 3 days prior to cutover • Collection and FTP of database delta's (archive logs) is ongoing from the point the backup is taken from test till closedown of Firm C's Test system for cutover • Integration testing required
Pre production	• Test environment testing is complete • Preproduction contingency database server build complete • Before testing a full database backup of Firm C Preproduction database is required • MQ migration solution in place & MQ Cluster scripts available from Firm E • Full database backup of Firm C Preproduction database is available 3 days prior to Cutover • Collection and FTP of database delta's (archive logs) is ongoing from the point the backup is taken from preproduction till closedown of Firm C's SCV Preproduction system for cutover • Integration testing • User Acceptance Test • Stress Performance Test • Backup/Restore Test • Disaster Recovery Test
Production	• Preproduction testing completed Before testing a full database backup of Firm C Preproduction database is required • Agreed MQ migration solution in place & MQ Cluster scripts available from Firm E • Full database backup of Firm C's Production database is available 3 days prior to cutover • Collection and shipping of database delta's (archive logs) is ongoing from the point the backup is taken from production till closedown of Firm C Production system for cutover • Production contingency database server build complete

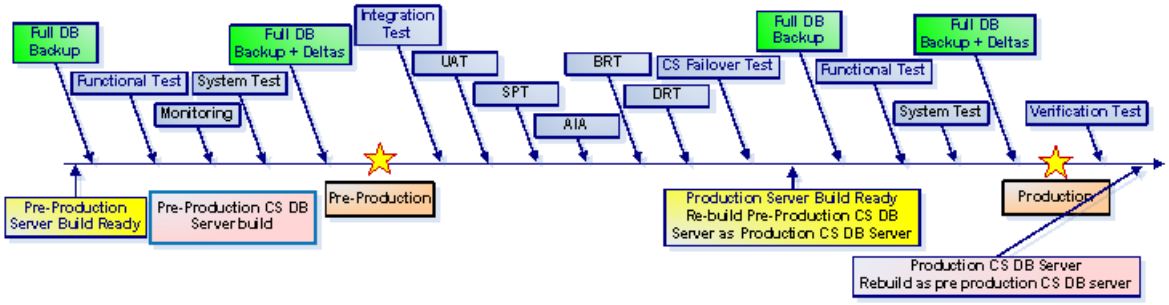
Migration timelines

Based on the above analysis of dependencies, the migration timeline is laid out in the solution plan, which is further used to develop the complete project schedule. Further, based on the days specified for the actual migration, migration implementation plans are drawn out giving minute by minute progress of the migration. These plans also carry details of the contact of each individual stakeholder who will participate in the migration process. Being quite detailed, these plans also carry information on risks and assumptions, which if not true, would result in a failed migration. The process to be followed during the migration is also laid out in these plans, alongside detailed perquisite tasks which are to be done prior to migration, and overall change plan steps which are to be done on the

day of the migration. A bridge line is set up on which the stakeholders can join in if some steps fails, and also attend one of 3 to 4 checkpoints kept at crucial stages of the implementation. An activity log and deployment log further takes care of any additional activities and deployment that may be necessitated during the course of the migration.



(a)



(b)

Figure 2: (a) Timeline till test environment migration
(b) Continuation of timeline to production migration

Communications management

Maintaining effective communication channels amongst stakeholders is one of the biggest challenges of the project manager in such a complex project where there is a lot at stake. Right from the design of the communications plan to effective management of all stakeholders, it is an art which comes through years of experience. Several challenges were identified in the management of effective communication throughout the life time of the project, viz.:

- More than 50 individual stakeholders spread across 6 countries
- Three time zones
- Obtaining sign off of deliverables by relevant stakeholders – some deliverables required as many as 30 iterative revisions before sign off
- Organizing weekly check points
- Setting up migration bridge line
- Urgent conference calls to resolve outstanding issues
- Managing and keeping track of the email surge
- Holiday and vacation calendar maintenance – ensuring appropriate backups
- Making schedule updates available on regular basis

A coordinated team effort with effective leadership allowed for generating strategies to overcome these challenges. Setting up separate mailing lists for system owners, client delivery unit, technical leads, technical teams, project

managers, etc. allowed for dissemination of information faster to the intended audience. A shared web portal, e-room, was made available for storing documents, plans, meeting minutes, application installation instructions etc. Training sessions were conducted by the technical architect and process managers to align processes and people to the specific needs of the project. By adjusting work shifts to have greater overlaps between teams working in different countries, efficiency in meeting timelines was increased to a point where very few deadlines were over shot. Documentation was a cornerstone of the entire effort as the ex-hosting vendor had not maintained any. A strict regimen and a common template meant the project manager was able to send out meeting minutes within a couple of hours of conducting a meeting. An issue/risk register was maintained and discussed in a programs level check point every week to mitigate any risks that arose in a timely fashion.

Effective change management process

A two tier change management process was called for, given the IS policy of Firm A, the owner firm and the in house process at Firm D, the hosting vendor. Each had its own change management tool, viz., Digital Workflow using Peregrine Service Center Client at Firm D and Focalpoint, provided by Telelogic at Firm A. The process at Firm D required a new release instance for each change on any given environment, while a single change request was sufficient for Firm A irrespective of the number of environments it was implemented on. A Change Advisory Board and a Technical Review Board was constituted to review and approve the change requests within a specified change request window, which was typically to draft the request, 9 days ahead of implementation for production environment and 3 days for non-production environments. Any changes which required expedient approval with short notice were classified into exceptional change requests and a separate approval process was followed for the same.

Monitoring Tools

The use of virtualized environments for the applications of Firm A was a new approach, and this called for suitable monitoring tools that would also build in reliability into providing uninterrupted fail safe service. Firm A was equipped with a custom design application that enabled viewing the business systems state of functioning at a high level on a web browser using red and green signals that indicated if a particular system was up or down. Additionally, CA monitoring was set up, which essentially is a performance monitoring product offering real-time transaction performance monitoring, performance trending, automated reporting, customized alerts and integration. Using the Opsware AMS agent, a framework for auto-updating the configuration management database was also set up, which meant that at any point of time, it would become easy for the monitoring tools to pin point errors in a particular configuration item leading to quick resolution of incidents. Over and above this, custom made scripts were written to enable database replication and sending out monitoring alerts relating to specific UNIX, DB and MQ components. Sending out these alerts requires auto generated emails, which needed special mail server permissions, and were a major unexpected bottleneck holding up sign off, until it was eventually resolved.

Table 3: Monitoring components

Architecture block	System components
Server O/S	• Hardware Mechanical Components • CPU Utilisation • Memory Utilisation • Disk I/O • File Systems size thresholds
SAN Storage	• Monitoring of the DMX SAN storage arrays and Brocade SAN Switches • Reporting using StorageScope
Middleware	• Database (Oracle, MS SQL*server) • Weblogic • MQ series
Network	• Router/Switches • LAN • Firewall
Application	• Process View

A HOLISTIC APPROACH FOR INTERRELATED SYSTEMS MIGRATION

Statistics reveal that the industry spends in excess of $ 5 bn per annum on data migration under several heads such as software, services, consulting, etc. With infrastructure requirements of corporates growing by leaps and bounds, this is expected to grow in terms of both capital invested and importance and sensitivity. However, it is also seen that such migration projects often overrun their budgets, get delayed or in certain cases, even get discarded. Part of the reason in this phenomenon is that the techniques and principles in data center migrations are not well understood, shoddily planned and executed in haste.

	2007	2008	2009	2010	2011	2012
Over Budget	562	619	681	749	824	906
Budget	5,055	5,560	6,116	6,728	7,401	8,141

Figure 3: Data migration cost forecasts and overruns *(Source: Bloor Research)*

One of the primary reasons behind the failure of migration projects is the lack of regard for experience and appropriate process selection required for data migration. By and large, data migration is often associated with applications. This is to say that it is often a part of some overarching applications project. This in turn, implies that very often, it is the applications folks who determine the migration strategy. However, it turns out that due their lack of expertise in dealing with data, the focus shifts to processes and interfaces rather than data semantics. Data issues are often ignored and as a result, the estimate for time, budget and personnel is incorrect leading to schedule and budget overruns. Very often, the application migration is detached from the data migration leading to further issues. What is thus needed is a clear focus on data management during the migration planning and execution. Further, the technical team needs to clearly understand the tools and techniques required for data migration.
However, in spite of all this, even with a certain focus on data migration, a number of issues typically crop up in such projects:

- Regarding data migration as a dull and unimportant activity
- Severe criticism for failure and few compliments for success
- Short term activity, after which, one may be back on the "bench"
- Priority to application team
- Software delivery delays leading to issues in planning data migration
- Tug of war between application and data migration teams
- Lack of co-operation from ex-hosting vendor
- Custom scripts built over lifetime of ex-hosting vendor
- Delays in obtaining sign off in certain deliverables from large number of stakeholders who don't agree on certain points

It is important to keep in mind that apart from the complexity that comes with inter-related systems migration, the

above issues further magnify the risks to the project. To address this problem, we can take some of the lessons learned in the successful implementation discussed in the case study and evolve some measures that help prepare a clear charter towards a successful datacenter migration effort.

Table 4: Inter-related Systems migration key strategies

Functional area	Strategy	Expected outcome
Architecture dependencies	• Lay out components of systems side by side • Generate DARS matrix to resolve dependencies	• List of Install packages and licenses required • Patch requirements defined • Order of migration steps
Environment dependencies	• List dependencies for each environment • Evolve solutions for each dependency	• Server build strategy • Testing strategy outline
Scheduling	• Create timelines for migration of each system • Organize program level meetings to ensure schedule compatibility between systems • Generate schedule based on agreed timelines • Baseline schedule at programs level	• Individual system schedules • Solution documents
Knowledge sharing	• Create shared web portal • Documentation of application install guidelines • Post minutes of meeting, system documents, capacity baselines, process documents, implementation plans on e-room	• Up to date system migration history • Reporting and tracking for monitoring and control
Monitoring and quality control	• Develop common monitoring strategy • Install monitoring tools, test send and receive systems alerts • Organize Weekly system checkpoints • Organize weekly program level meetings	• System status updates • Issue/Risk register updates
HR and communications management	• Proactively reward and acknowledge success in achieving key milestones • Identify key members for higher roles in new projects within organization post successful migration • Greater focus on data migration vis a vis application migration • Clear and concise communication with ex-hosting vendor – focus on collaborative team efforts • Strong and experienced technical team to build custom scripts • Priority to sign off requests with pre assigned deadlines	• Increased morale in team • Clearly defined data migration steps and minute by minute implementation plan • Increased co-operation from ex-hosting vendor • Higher completion status of Server builds • Higher completion status of signed off deliverables

CONCLUSION

Data center migration of multiple inter-related systems is an extremely complex activity. In essence, it is a discipline in itself, calling for the maximum one could stretch one's imagination in terms of collaboration, coordination, co-operation and team work. The presence of multiple corporate entities further complicates matters in many projects, requiring new ideas for effective management in all the knowledge areas of the project management framework. There is a high risk of failure, and achieving the triple objectives of scope, time and cost can be quite a challenge. In this background, it is only a pre planned holistic approach keeping all these factors in mind, that can steer the course clear for successful execution of the projects. Understanding major dependencies and evolving suitable solutions that resolve these dependencies is crucial to obtaining a head start in such projects. However, this

alone is not enough and should be backed up with compatible scheduling, regular updates, keeping all communication lines open, maintaining high team morale, and tracking progress throughout, keeping all relevant stakeholders in mind at all times. Adopting a comprehensive strategy towards management of such projects can go a long way to ensure fool proofing of such projects from failures and budget and schedule over-runs, with improved profitability from such large scale efforts.

REFERENCES

"A Guide to the Project Management Body of Knowledge PMBOK® Guide—Fourth Edition" (2008), *Project Management Institute*

Philip Howard & Carl Potter (2007), "Data Migration in the Global 2000", *Bloor Research*, Available online at http://www.bloor-research.com/research/survey/876/data-migration-survey.html

Anonymous (2007), Intel Solution Services, Facilitating a Transparent Data Center Migration, Available online at http://cache-www.intel.com/cd/00/00/36/90/369009_369009.pdf

Anonymous (2007), "Data Center Migration: 4 Models for IT Integration", Available online at http://www.podtech.net/home/3144/data-center-migration-4-models-for-it-integration

ABOUT THE AUTHOR

Amar Kumar Behera is a member of the PPMG (Project & Program Management) group at Mphasis an EDS company, since August 2008. He is involved in infrastructure management projects, lean six sigma quality control, design for IT service management and managing transitions for outsourced managed services.

He holds a B.Tech. in Manufacturing Science & Engineering, a M.Tech. in Industrial Engineering & Management and Minor in Electronics & Electrical Communication Engineering from the Indian Institute of Technology, Kharagpur. He also holds a M.S. degree from the College of Engineering at the University of Illinois at Urbana-Champaign. Besides, he is also certified in ITIL V3 (Information Technology Infrastructure Library) Foundation given by EXIN, Netherlands. Recipient of the C J Gauthier Fellowship and Director - IIT Kharagpur's letter of special mention, he has published 10 articles in peer reviewed conferences and journals thus far.