Developing a methodology for large IT project migration and integration in case of acquisitions, mergers, that are critically constrained by time and budget

Sanjana Shashi Kadapa

***Abstract*- Migration of large IT projects becomes complex, especially where a large number of software assets are involved. IT migration is essential to integrate legacy systems with new systems, and when large firms are acquired by another firm. A structured methodology is needed for seamless integration. This paper provides a literature review of different models used for IT migration, and recommends a model that can be used in complex case scenarios.**

***Index Terms*-** IT migration, systems integration, legacy migration, migration models

1. INTRODUCTION

IT migration projects in the manufacturing industry involve not only databases and application migration but IT infrastructures systems. Typically, when a large firm is acquired or merges with another, then migration involves full IT infrastructure and supporting structures of process control, quality systems, VoIP systems, manufacturing systems and many other critical data and information systems (Cohen, 2009). As an example, when Ford and Jaguar Land Rover – JLR merged the IT managers realized that both firms had decades of international experience and proven systems, and these could not be set aside. Unless a systematic framework and method is adopted, the process of migration can become a nightmare. This paper reviews different methods available for migration and examines how these methods can be integrated to form a common framework for migration. A model has been created with inputs from various models that can be used to address the risks and opportunities for the problems identified.

1.1 The need to Migrate IT projects

In today's business environment, mergers and acquisitions have become a reality and worth hundreds of millions of pounds. Unless the IT systems of these firms are migrated to a common system, the business will not be to exchange data, speak with each other, cannot use each other's synergies and would become silos of information. IT systems would have the real intellectual capital of the firms and information about clients, bank, financial and operational data, employee database, product database, stores and inventories, production systems and many other areas (Cohen, 2009). With migration, information from one firm can be moved to the IT system of another firm or in some cases, a middleware is used that connect the applications from one company to another. Some firms may even have ERP systems from different vendors such as SAP, BAAN and JD Edwards and data has to be integrated from one system to another. Hence, migration ensures business continuity and that the intellectual property is not lost after a merger (Ulkuniemi, 2002).

1.2. About migration in the case of acquisitions and mergers

A review of the literature on migration, by the author of this paper shows that in the case of mergers and acquisitions, there would be at least two entities one is the acquiring firm called the acquirer and the other is the target firm. These firms would have been functioning independently for a number of years and have their own IT systems. Two scenarios for migration, Scenario 1 and Scenario 2 are given as follows.

The following two scenarios arise in migration:

- Scenario 1: The firm for whom migration has to be done has a number of standalone legacy applications that are being used by different departments and which have not been integrated before. This is the worst possible scenario and vetting out the data, analyzing the systems can take a few months besides requiring a number of skilled resources.

- Scenario 2: The firm already has an integrated ERP system and in this case the merging and integration can be done with the least expense and time.

The author of this paper would take the perspective that migration of large projects is best done with a project management approach. A large migration project would have a number of milestones, deliverables, stakeholders and resources. Resources include the system engineers, end users from departments such as manufacturing or marketing, etc. and the appropriate hardware and software. By adopting a project management perspective, it is possible to act within the time and cost constraints.

2.2. Information that is Migrated

The term migration and integration refer to specific applications in the migration scenarios that have to be migrated. There are a number of operating systems with different architectures for the applications. These applications would be coded in different languages and they would have different user interfaces and different types of datasets. Following figure shows some of the assets that can be migrated.

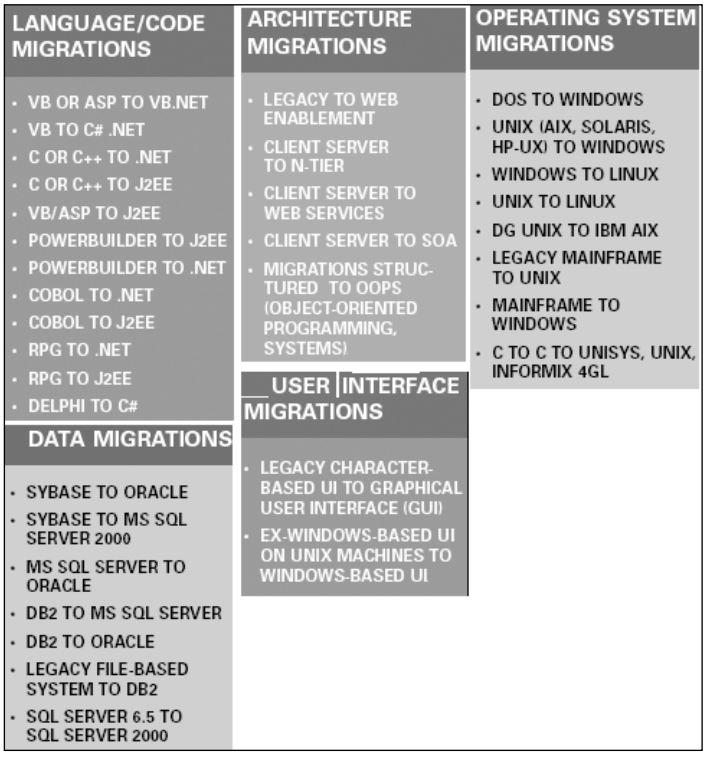


Figure 2.4 Assets that can be migrated (Wu, 2007)

A migration exercise can involve one or more of the more activities that are indicated in the above figure. When migration is done at the database level or at the operating system and the architecture level, then the cost and schedule is more. When it is to be done at the user interface level or at the language and code level, then the efforts are lesser.

2. Literature Review – Migration Methodologies and Frameworks

The literature review would attempt to cover the areas that are illustrated in the above figure. This type of literature review would ensure that the current business process and strategy are analyzed along with a selection of the strategy and architecture. Migration would also involve a change management approach and this should be evident in the review. In the next sections, some popular methodologies and frameworks are briefly discussed. A critique is provided for each method and the decision of either Recommended or Not Recommended is given. In some cases, a decision of Partly Recommended is also suggested.

Migration methodologies and frameworks would depend on the extent of migration to be done. It should also be noted that migration is not related to only technology issues but a change management approach has to be adopted and there are issues related to HR,

process management and strategic management. Migration can be regarded as an opportunity to improve both the organization processes (Teppe, 2009). The following figure illustrates the process model for migration.

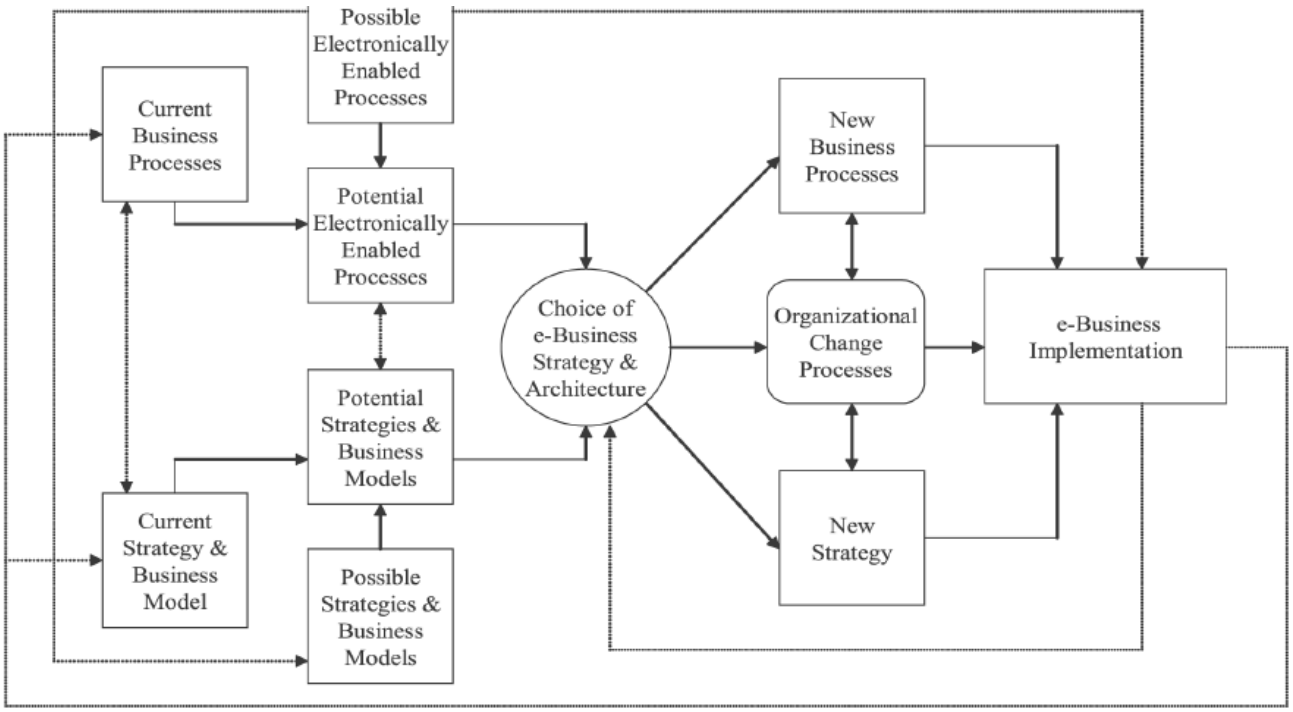


Figure 2.1. Process model for migration (Krell, 2005)

2.1. Types of Migration

Some common types of migration are:

Complete Migration: In this type, all the assets are migrated and intermediate validation of the business rules is not allowed. The strategy is risk prone and prior evaluation is not possible.

Iterative Migration: Migration is done as per the modules and components required and thus there is greater control over costs and time. However, the application has to be well organized as per modules.

Limited Migration: Only one component is selected and migrated. The component may also be improved and upgraded as per requirement.

Vertical Migration: Migration is done by replacing each tier of application through all the tiers. However, the method is possible when each tier of the application is isolated and would have a minimal impact on the system.

Horizontal Migration: An entire tier is migrated without migrating other tiers. This is possible where applications have large quantities of servers, shared code and complex middle tiers

2.2. SDM/ PPM methodology at Ford and JLR

Ford and JLR have developed an integrated solution that combines two discrete models, project management methodology to the solution delivery model called the SDM/ PPM model. These two models are discrete and different but they are integrated so that project management techniques do not remain a theory but also tell the project manager as to what has to be done. (One IT PPM Team, 2004).

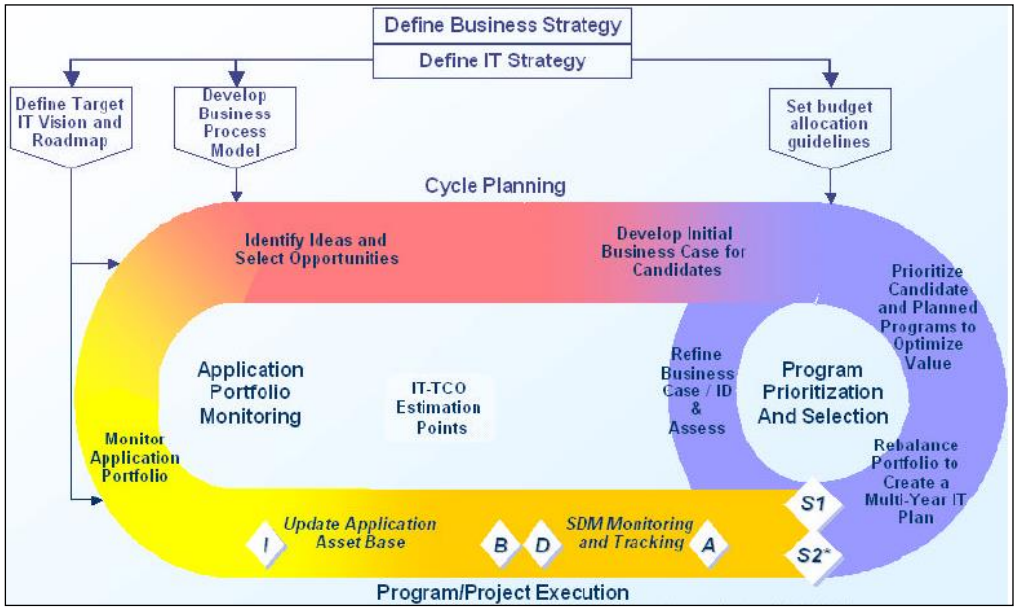


Figure 2.2. SDM/ PPM Model at Ford (One IT PPM Team, 2004)

Critique: This method has to be strictly monitored by ensuring that software improvement and upgrades do not creep into the project. Overall, it can be combined with the six-sigma method and other methods for migrating manufacturing projects. Recommended.

2.3. Prince 2 Model

Projects IN Controlled Environments method allows project management as per a defined framework. The method specifies the procedures used to carry out coordination between activities and people involved in the project. It helps to carry out the project design and supervision and offers contingencies for making changes if required. Each process would have important inputs and outputs with a number of activities required to carry them out. The method allows for the control and management of resources efficiently (Udechukwu, 2007).

Critique: The model is commendable when the framework is clearly defined and scope creep is reduced. However, in large migration projects, unknown variables are a reality and hence the model is inappropriate. Not Recommended.

2.4. Six Sigma Model

The methodology attempts to align problem solvers to Customer needs and Business Goals driven from the top, improves business measurements and competitiveness. It is a step-by-step approach to reducing variation, increasing quality, customer satisfaction, and, in time, market share. The methodology is based on six phases termed as DMAIC that stands for Define, measure, analyze, improve and control. There are twelve steps that make up the phases and each phase has a set of steps that have to run sequentially.

Critique: The method is best used along with RUP or the SDM/ PPM method. It is most appropriate for manufacturing environment where rejections and rework become very expensive. Recommended.

2.5. Waterfall Method

The method is used mainly for development projects where each step flows like a waterfall. There are different steps such as requirements, design, implementation, verification and maintenance. It was used in the manufacturing and construction industry and requires that each step must be perfected before the next one can commence. A very high reliability can be achieved with this method (McConnell, 2006).

Critique: In large IT migration projects, unknown variables and factors emerge and must be accommodated in the project plan. Waterfall does not all rectification and modification of previous steps. Hence, this model is not suitable. Not Recommended.

2.6. Agile Method

Agile methodology is uses elements of iterative development and cooperation with cross-functional teams. When the teams are highly motivated, well adapted with high teamwork and accountability, then the method can be used. The method believes in creating small functional units that can be developed quickly (Cohen, 2004).

Critique: While the method is highly flexible and allows for quick development, it cannot be used for the whole project since integration would be a major problem. It can be used for small modules and functionalities of the system. Recommended Partly.

2.7. Iterative Method

The iterative method is the base methodology and framework from which some other models such as RUP and agile have been developed. The method is cyclic in structure and starts with the initial planning and ending in the final deployment. The method allows teams members to learn from past mistakes and then develop more mature processes (Fujita, 2003).

Critique: The method while allowing teams to learn and develop skills cannot be considered for migration projects that are constrained by time and budget. Hence this method is considered more as a guiding philosophy rather than as the main framework. Recommended Partly.

2.8. SCRUM Method

SCRUM is used in smaller software development process when targets are clearly defined and the variables are controlled. In such cases, different phases of the project would overlap and small teams would handle each phases, guiding and handholding till the next phase takes shape (Fujita, 2003).

Critique: The method cannot be used for large complex IT migration projects since the deliverables are many and variables often uncontrolled. Not Recommended.

2.9. RUP Method

The Rational Unified Process was introduced by IBM for the software industry where iterative development of software was. Different elements in the methodology will be selected by the project teams as per their requirements. There are different phases in the framework and they are inception, elaboration, construction and transition (Fujita, 2003).

Critique: The method has been adopted in the methodology adopted by Ford and JLR. Recommended.

2.10. Six Step Migration Methodology

The migration process is typically run as a project management with a structured set of steps. Following figure, show the recommended six step process (Harriman, 2004).

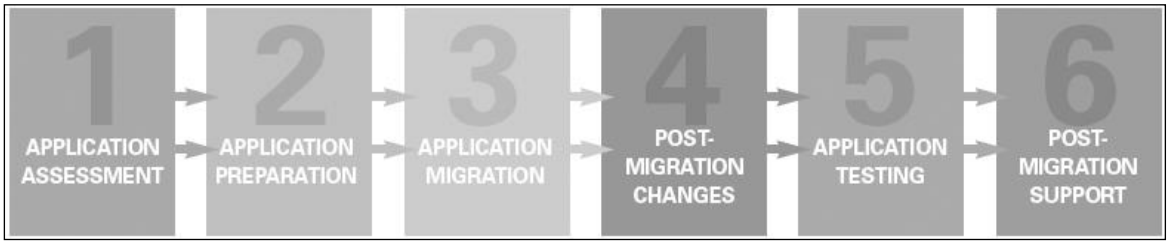


Figure 2.3. Six Step Migration methodology (Harriman. 2004)

Critique: The method is best used when migration is to be done at the application level. It can be used for migration of applications such as SCM, Finance, and HRM. Recommended Partly.

2.11. SOA and Migration for Cost and Time Constraints

The author of this paper comments after an extensive literature review about the emergence of SOA as the best method to control costs and budgets. SOA is an architectural pattern that says that computational units such as system modules should be loosely coupled through their service interfaces for delivering the desired functionality. Following figure shows the typical SOA interactions.

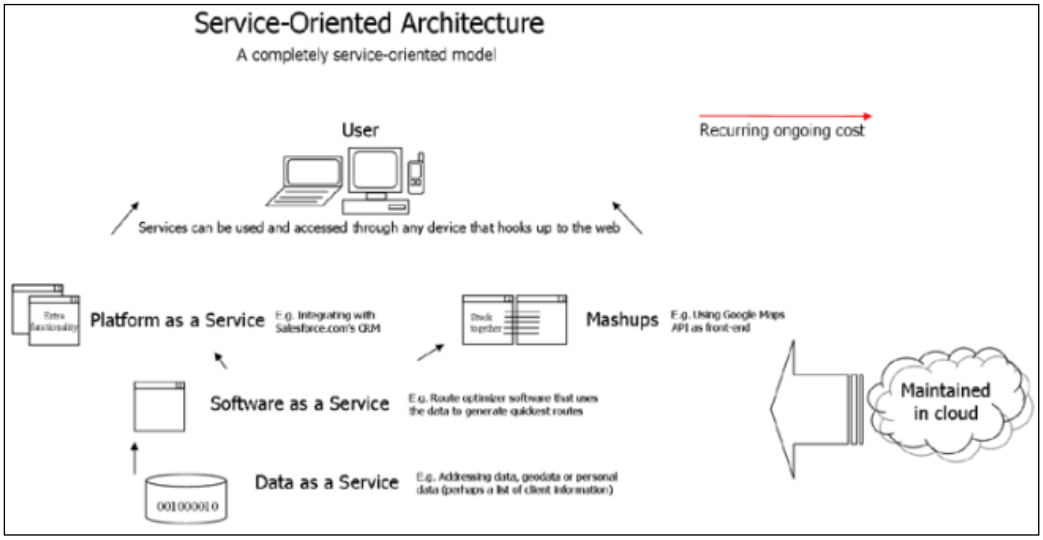


Figure 2.4. Typical SOA Interactions (Peng, 2008)

A review of the literature by the author of this paper shows that the services can be based on any platforms such as Corba, Jini, Web Services, and Mainframes and so on. With the introduction of enterprise application integration tools, firms have addressed the challenges of migration extremely well. Vendors such as webMethods, Tibco, and SeeBeyond provide enterprise application integration - EAI tools that can connect packaged applications and custom applications across the enterprise using either a single bus or a hub for all kinds of integration needs. The challenge for migration to be effective is that down to the tables and row levels, the system should know where to write what data. With SOA, for example, an employee information service with operations such as getContactDetails, getPersonalDetails, or searchEmployeeByLastName can act as a single source for employee related data (Aversano, 2001). Following figure shows the functional layer concepts in SOA.

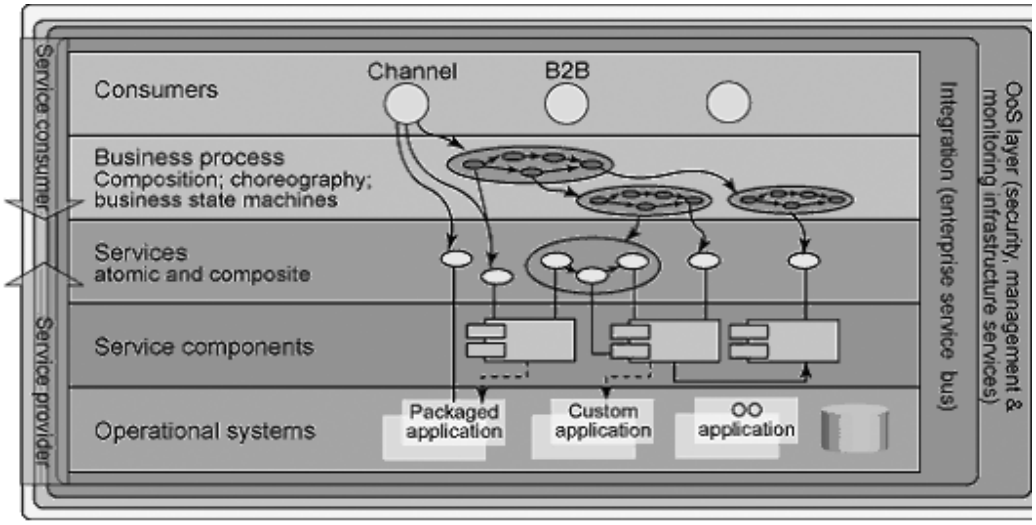


Figure 2.5. Functional layers in SOA (Portier, 2007)

As seen in the above figure, the SOA would have a number of functional layers. The legacy system to be migrated and the new application to which migration takes place can be placed in different layers. The two systems can be connected by middleware or

service components and help users from both systems to retrieve data. SOA is more cost effective both in terms of time and money and it can be considered as a suitable alternative for enterprise integration.

2.1. Six Step Migration Methodology

The migration process can be split into six steps and these are briefly explained as below: Step 1- Application assessment: The application and the modules to be migrated is assessed for total cost of ownership, cost to benefit ratio, if migrating and carrying forward the module would be useful or no. The following questions have to be answered in this step.

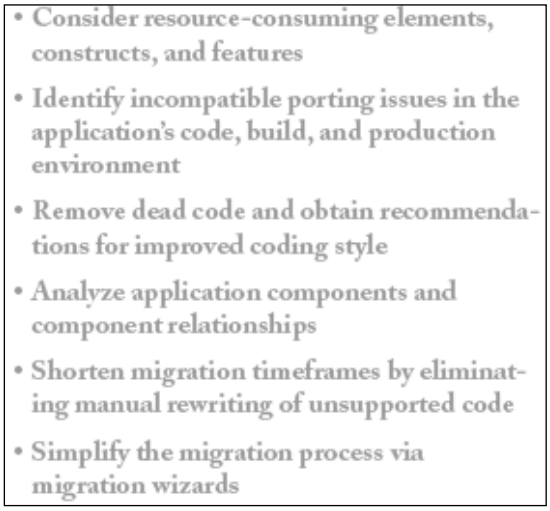
- What functionality does the application possess that other applications or third-party tools cannot reproduce?
- What types of data and data transmission protocols does the application support?
- What are the application's basic input and output types, different interface points, and external dependencies?
- Does the application handle legacy file formats or high value business transactions?
- How would removal of the application impact the organization?
- What is the current TCO for the application? Would TCO improve if the application was ported to the new environment?
- What is the application's size? How many lines of code, forms, user controls, modules, classes, and data source types exist?
- What are the application's functions, properties, and types?
- How complex is the application? What application features are not supported, resulting in a potential manual migration?
- Does the application depend on other tools or use an internal mapping that generates internal functional dependencies?
- Is the application currently undergoing enhancements or code changes?

This stage becomes very important as the complexities, costs and time required for the migration can be estimated. With these factors known, the efforts estimation becomes more accurate and hence the management can then decide whether to accept the cost and time required for the migration or to consider alternatives.

Step 2 - Application preparation: The second step is to prepare the application for migration. Some initial conditions have to be met before the migration starts and these are:

- Provide all the relevant application documents and baselined source code to the migration team
- Supply functional experts to the migration team for accurate understanding of the project
- Provide application source code that has not undergone separate enhancements as the migration begins

Step 3 - Application migration: Migration commences in this step and done by using migration tools. By using such tools, overwriting, code replacement and other actions are done automatically. For areas where the code was not migrated automatically, manual intervention has to be done. Migration tools have some advantages as given below:



A migration status report is usually generated to show where the code has not been updated automatically.

Step 4 - Post Migration Changes: In this step, code that has not been automatically updated is migrated manually. In some cases, manual rewriting of code is required to meet new objectives of the migration exercise and this effort can be substantial.

Step 5 - Application Testing: The migrated application is subjected to severe tests such as functional testing, load, volume and stress testing and other types of testing are carried out to ensure that the migrated application would perform as required. If required enhancements and upgrades can be suggested.

Step 6- Post Migration Support: Issues that arise when new application is used for some time are sorted out by the support team. There may be some bugs, errors, upgrades and other factor that have to be sorted out.

3. Critical Analysis and Discussions

In the previous chapter, a number of methods have been presented along with the justified comments of Recommended, Not Recommended and Partly Recommended. Methods that are not recommended are ignored in this chapter. The recommended methods are SDM/ PPM methodology at Ford and JLR; Six Sigma and Rational Unified Process. Methods that are Partly Recommended are Agile Method; Iterative method and Six Step Migration Methodology. The existing methods cannot be used as standalone models but they have to be merged with other models. A new model is recommended in this section, based on the above methods.

3.1. A New Migration Methodology

There is sufficient published literature on constructing and evaluating business models. Osterwalder (2005) says, “A business model is a conceptual tool containing a set of objects, concepts and their relationships with the objective to express the business logic of a specific firm”. Magretta (2002) suggests that "a good business model begins with an insight into human motivations and ends in a rich stream of profits". Shafer (2005) suggests that due care should be proposed while proposing a business model as the model would be applied to conditions and events that the author may not have considered. Based on the literature review, a business model is proposed for migration in joint ventures between two or more firms. Following figure illustrates the model.

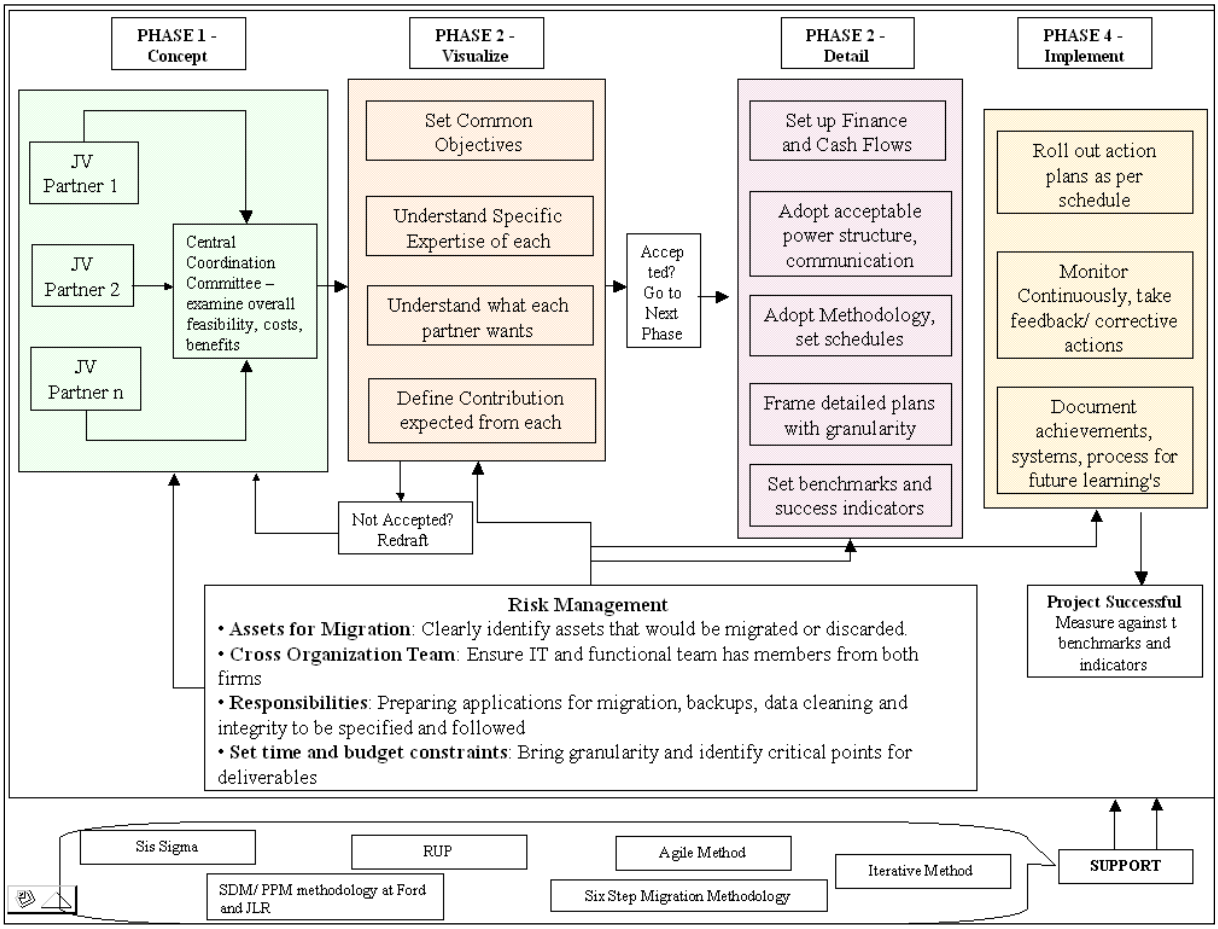


Figure 3.1. Developing a New Model for Migration (Authors Construct)

As seen in the above model, there are four phases for a JV and a risk management phase that is active through the whole project life.

Phase 1 Concept: In this phase, the project concept is framed and two or more partners and stakeholder have merged or acquired. The concept of migration of systems would begin at this stage.

Phase 2. Visualize: In this phase, activities such as setting common objectives, understanding what each partner wants and contribution expected from each other. Once these are drafted, then they have to be approved else the specifics have to be redrafted.

Phase 3. Detail: In this phase the JV details are set to granularity. Details of finance, cash flows, power structure, communication systems have to be set up. Next methodology, detailed plans have to be framed along with schedule and timelines. Next benchmarks and success indicators have to be defined to judge if the project is successful.

Phase 4. Implement: This is the implementation phase where the action plan is rolled out. There has to be continuous monitoring and feedback with corrective actions. Documentation of achievements, systems, process has to be done for future learning's. If the project meets the benchmarks of success indicators, then the project is deemed successful.

Risk Management: This process would occur through the life of project and various risks are identified. By using an open and transparent mode of communication, by sharing information and power, distrust can be avoided between the partners.
Support: A number of methodologies that were reviewed are used as support systems for the migration project on a need basis.

4. CONCLUSIONS

The paper has examined various methods and frameworks that could be used for migrating large projects with special significance in the manufacturing industry. A number of models are available and the review provided a brief list of some models. However, these models could not be directly used for migration since they were centered on software development and maintenance projects. Moreover, the models did not consider the risks associated with mergers and acquisitions. Important aspects were selected from each model and then a new model has been proposed that would address the shortcomings of all other models. The new model has four phases and considers risk management and also takes support from other existing models. The new model could be tested on pilot study basis and amended as required.

REFERENCES

1. Aversano. Lerina, 2001. Migrating Legacy Systems to the Web: an Experience Report. Proceedings of the Fifth European Conference on Software Maintenance and Reengineering, IEEE Proceedings Publications, pp. 1-10
2. Cohen, D., Lindvall, M., 2004. An introduction to agile methods. In Advances in Computers. New York: Elsevier Science.
3. Cohen. Stephen, 2009. Service Migration in Enterprise System Architecture. Proceedings of the 42nd Hawaii International Conference on System Sciences - 2009, IEEE Proceedings Publications, pp. 1-10.
4. Fujita. Hamido., 2003. New Trends in Software Methodologies, Tools and Techniques. Volume 98 Frontiers in Artificial Intelligence and Applications, Amsterdam: IOS Press
5. Harriman. Alan, 2004. Emergent Database Design: Liberating Database Development with Agile Practices. Proceedings of the Agile Development Conference, IEEE Proceedings Publications, pp. 1-8.
6. Heymans. Louis, 2007. Testing Techniques for the Cross-platform Migration of Very Large Interactive Applications. 11th European Conference on Software Maintenance and Reengineering, IEEE Proceedings Publications, pp. 1-2.
7. Krell. Terence, E-business migration: a process model. 2005. Journal of Organizational Change Management, 18(2), pp. 117-131.
8. Magretta. Joan., 2002. Why business models matter. Harvard Business Review, Harvard Press, pp. 4-8.
9. McConnell. Steve., 2006. Software Estimation: Demystifying the Black Art. Microsoft Press.
10. One IT PPM Team, 30 April 2004. Project Management Process Guide Production Version 2.3. Ford Internal Proprietary Document, Ford, USA.
11. Osterwalder. Alexander., May 2005. Clarifying Business Models: Origins, Present, and Future of the Concept. Communications of AIS, 15, pp. 1-43.
12. Patel. Shushma, & Kirk. Julie, 2001. A Methodology for Migrating a Client/Server Application to an Intranet. Proceedings of the 39th Int'l Conf. and Exhibition on Technology of Object-Oriented Languages and Systems, IEEE Proceedings Publications, pp. 1-13.
13. Peng. Kuang-Yu, & Lui. Shao-Chen., 2008. A Study of Design and Implementation on SOA Governance, A Service Oriented Monitoring and Alarming Perspective. IEEE Computer Society, 978-0-7695-3499-2/08, pp. 215-221.
14. Portier. Bertrand., May 2007. SOA terminology overview, Part 1: Service, architecture, governance, and business terms. Retrieved 18 September 2009 from http://www.ibm.com/developerworks/webservices/library/ws-soa-term1/
15. Shafer. Scott. M., 2005. The power of business models. Business Horizons, 48(3), pp. 199-207
16. Teppe. Werner, 2009. The ARNO Project: Challenges and Experiences in a Large-scale Industrial Software Migration Project. European Conference on Software Maintenance and Reengineering, IEEE Proceedings Publications, pp. 149-158.
17. Udechukwu. Ojiako., & Johansen. Eric., December 2007. A qualitative re-construction of project measurement criteria. Industrial Management & Data Systems, 108(3), pp: 405-417
18. Ulkuniemi. Pauliina & Seppänen. Veikko, 2002. Definition of a COTS Software Component Acquisition Process Framework: The Case of a Telecommunications Company. Proceedings of the 28th Euromicro Conference, IEEE Proceedings Publications, pp. 1-9
19. Vosburg. Jodi, 2001. Managing dirty data in organisations using ERP: lessons from a case study. Industrial Management & Data Systems, 101(1), pp. 21-31.
20. Wu. Bing, 2007. Legacy Systems Migration - A Method and its Tool-kit Framework. Proceedings of the 4th Asia-Pacific Software Engineering and International Computer Science Conference, IEEE Proceedings Publications, pp. 312-320.

AUTHORS

First Author – Sanjana Shashi Kadapa, Bachelor of Engineering (Computer Science), Cummins College of Engineering for Women, sanjanakadapa@gmail.com.

	After completing her Bachelor of Engineering in Computer Engineering from Cummins College of Engineering for Women, Pune University, Sanjana took up community service and taught several underprivileged children about computers and web technology. Since August 2015, she works as a software developer with HSBC - Global Technology Center with the Data warehousing practice. She is planning for a MS in computer science in USA.

Correspondence Author – Sanjana Shashi Kadapa, sanjanakadapa@gmail.com