Software Engineering Challenges of Migration Projects

S.Geetha, Assistant Professor, Dept.of Computer Science
Madurai Kamaraj University College
Email:geeta_baskar@yahoo.co.in

Dr.Alagaramy, Associate Professor, Dept.of Computer Science
Madurai Kamaraj University

Abstract

Organisations often face the challenges of migration from legacy systems to new target systems. Such migration efforts represent a complex engineering problem. This paper describes migration planning, identifies influencing factors, outlines a set of migration planning activities and offers a set of guidelines for the migration planning.

Keywords: migration, planning, legacy, strategy, process

1. Introduction

Many organizations face the problem of having to modernize their existing software systems by migrating to more capable systems. Modernizing legacy software systems is a complex engineering problem that includes most aspects of traditional software development with more constraints. A successful migration effort requires both a sound development plan and a sound migration plan.

2. Migration Strategy

Migration can be instituted across a range of technology classes, including the following:

- Language or code migrations
- Operating system migrations
- Data migrations
- User interface (UI) migrations
- Architecture migrations, including migration to object-oriented programming (OOP)

As a result, enterprises can perform migrations whenever a greatly improved infrastructure is desired, including programming languages, operating systems, data, architecture, or any combination of these. Organizations determine which of these strategies to use based on factors such as system qualities, manageability, training, and cost.
2. Migration Process

Migration redevelopment is unacceptably risky and wrapping is unsuitable, migrating the legacy to an open environment can be the best alternative. Although it is much more complex undertaking than wrapping, if successful migration’s long-term benefits are also greater. Migration offers more flexibility, better system understanding, and easier maintenance and reduced costs.

A good migration plan should weigh programmatic and technical drivers for system development against customer priorities. Because of this, the plan may impact system development and certainly should impact system deployment. Iteration among the key stakeholders is necessary for an effective migration effort. Like development, migration planning involves tradeoffs among cost, schedule, risk and resources. Complex mainframe applications are unable to keep pace with, and respond to, today’s dynamic business demands.

This presents several issues, including:
- Increased maintenance costs of current IT environment
- Limited resources with specialized skill sets
- Lack of interoperability among disparate systems
- Lack of scalability and functionality in the current system
- Lack of agility to incorporate new features rapidly

Migration sets the stage as a beneficial opportunity to move these old mission-critical applications into the 21st century and resolve these issues.

4. Migration Planning

The following figure is based on experience with reengineering and migration efforts. The activities that are part of developing a migration plan.

**Figure 1 Scope of Migration Planning Activities.**
Migration planning relies on planning and development artifacts as inputs. The inputs should include the following information:

- organization’s goals and objectives
- priorities and user needs
- technical approach
- available funding and resources
- functionality of systems/products
- non-functional requirements (performance, security, interoperability)
- relevant regulations, policies, standards, and business rules/doctrines
- user manuals and training aids describing legacy system capabilities

4.1. Describe the Migration Management Approach

Managing the migration effort is a critical success factor. It is important to determine how ongoing enhancements to legacy systems will be managed while the target system is being developed.

The migration management approach needs to consider how to:

- Track progress of milestones.
- Identify and monitor open issues.
- Identify and mitigate risks.
- Establish communications among system developers, points of contact for the legacy systems, organizations responsible for interfacing systems, and customer and user groups.

4.1.1. Identify the Migration Activities

Prototypes can effectively test the potential solution, especially in cases where current systems are complex and involve many users. The migration plan should identify prototyping needs. This activity also should address the scope of the prototyping need, the migration concepts that are being tested, a set of expected outcomes, and mechanisms to evaluate whether the expected outcomes have been achieved. The prototypes need to be meaningful, and they need to be more than just a public relations type of “demo”. Prototype solutions can be evaluated through a variety of means, including proof-of-concept evaluations, user evaluations, and architecture evaluations.

4.1.2. Approach for Completion of Migration

After each new rollout and training phase is complete, the developer needs to ensure that users migrate to the new target system. Basic issues include both the timing and the extent to which users will be able to convert to the new system.

The migration plan should develop approaches to accelerate the migration effort and to decommission the legacy systems as soon as practical. Extra effort may be necessary to accommodate the “late adopters” and other users who are experiencing unanticipated or thorny problems. This may require developing temporary workarounds until a suitable solution can be developed, tested, and included in the next software release or system rollout. Another aspect of this activity is estimating the time and cost to complete the transition of all users to the new system and to decommission the old systems. The developer should consider incorporating the following approaches to help migrate users of legacy systems:

- Inform each group of users how and when they should transition their specific tasks and workload from their legacy systems.
• Establish incentives to fully transition operations and eliminate dependencies on legacy system operations.
• Provide assistance or software utilities for converting legacy databases.

5. Migration Planning Process

Previous experience has identified checklists for system evolution, enumerated reasons why reengineering projects fail, and identified guidelines for migration efforts.

These guidelines include:

1. Analyze the needs of the affected stakeholders to determine migration schedules, training requirements, and operational cutover to the new system.
2. Develop quantifiable measures of success for the migration effort.
3. Initiate the migration planning effort at the outset of the project before the development and implementation approach is set in concrete. Clearly define lines of communication and authority and provide adequate resources. Do not treat the effort as an added task for assigned team members.
4. Involve customers and users in the migration planning effort.
5. Do not allow system implementation to begin until a migration plan is approved and the “buy-in” of the affected stakeholders is obtained.

Conclusion

Despite its importance to success, the development of migration plans has often been neglected in the past. Nevertheless, a migration plan can help ensure that a development organization can successfully transition an active user community from a legacy system to its replacement. A realistic migration plan addresses the deployment and transition issues associated with migrating to a new system and phasing out an existing system. This white paper outlines the activities involved in migration planning. It presents steps that an organization can take in developing a migration plan. It also provides a set of guidelines for implementing a migration plan.

References


[2]. Software Engineering. K.K.Aggarwal


