DoD Software Migration Planning

John Bergey
Liam O'Brien
Dennis Smith

August 2001

Product Line Practice Initiative

Technical Note
CMU/SEI-2001-TN-012

Contents

List of Figures

Abstract

The DoD often faces the challenge of migrating from legacy systems to new target systems. Such migration efforts represent a complex engineering problem. These efforts call for a migration plan to supplement the development plan. The migration plan addresses issues associated with phasing out legacy systems and moving to the new system. These issues include user interface compatibility, database compatibility, transition support, system interface compatibility, and training. By producing and implementing a migration plan, a development organization can help a user community make the transition in an orderly fashion.

This note describes migration planning, identifies influencing factors, outlines a set of migration planning activities, and offers a set of guidelines for the migration planning process.

1 Introduction

Many DoD 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 [Feiler 93]. A successful migration effort requires both a sound development plan and a sound migration plan.

A development plan addresses the selection of the appropriate software processes, methods, tools, and hardware and software platforms. A migration plan created in conjunction with the development plan is necessary to help ensure that the operational transition to the new system goes smoothly.

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.

Migration issues have been identified previously [Bergey 97, Bergey 99a]. This current note focuses on migration planning, identifies influencing factors, outlines a set of migration planning activities, and offers a set of guidelines for the migration planning process.

2 Migration Planning

Figure 1 is based on experience with reengineering and migration efforts [Bergey 97, Bergey 99a]. This figure illustrates the activities that are part of developing a migration plan.
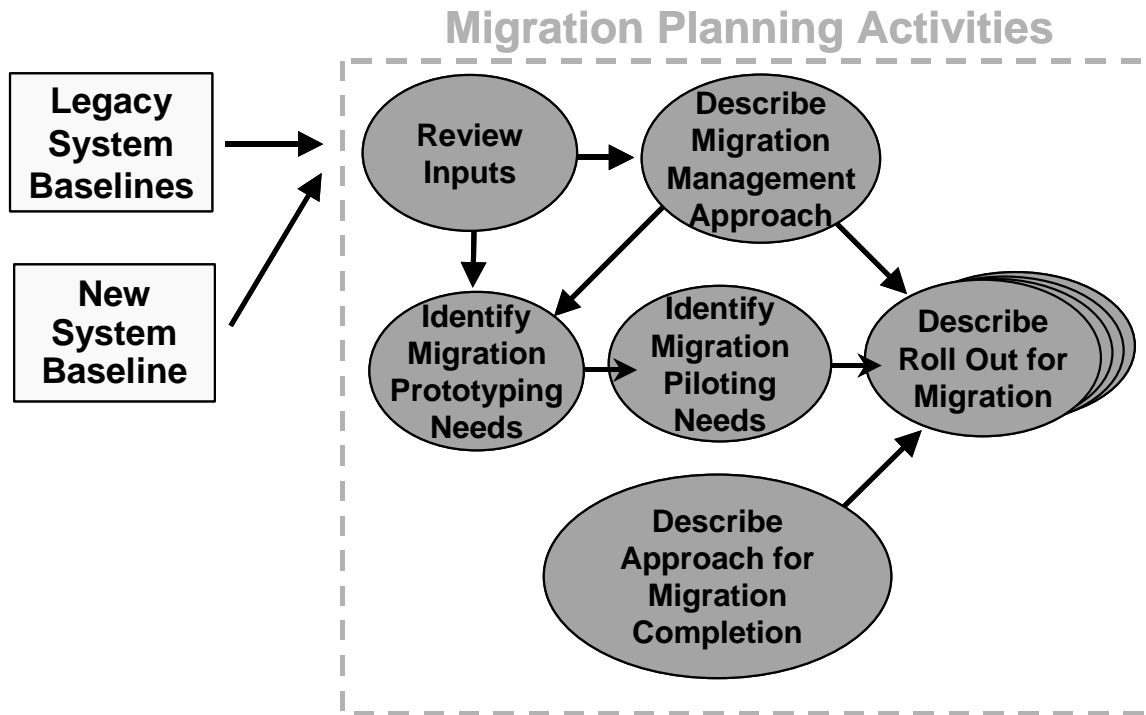
Figure 1:   Scope of Migration Planning Activies

These activities are outlined in the following subsections.

2.1 Review Inputs

As shown in Figure 1, migration planning relies on planning and development artifacts as inputs. These inputs provide a baseline description of the current state of the legacy systems, a baseline description of the new system, and a concept of operations that describes the target system from a user-oriented point of view. A review of these artifacts provides an initial understanding of the degree of compatibility between the new and legacy systems.

The inputs should include the following information:

- organization's goals and objectives
- customer 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

Using this data, the review of inputs should address the following set of questions:

1. What are the major issues that are likely to occur during the transition period?
2. What are the customer priorities and user needs?
3. Will the user interface be compatible, including inputs and outputs, reports and databases, and files that users depend on to perform their job responsibilities?
4. What are the practical constraints on funding and resources?
5. How do the funding constraints affect the organization's ability to satisfy its goals and objectives?
6. Who are the specific customers, and what are products and services that are to be supported by the new system? Who are the specific customers, and what are products and services that will not be supported?
7. Is there a need to support overlapping operations between the legacy systems and the new system? For how long?
8. Are the target system databases compatible with existing legacy databases?
9. Is there a need for software utilities for converting existing legacy databases?
10. What is the ability of target system interfaces to support existing legacy system interfaces with other external systems?

2.2   Describe Migration Management Approach

Managing the migration effort is a critical success factor. It is important to determine how on-going 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 on 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.

2.3 Identify Migration Prototyping Needs

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. At the same time, it should address the extent to which migration considerations call for prototyping both to mitigate risks and to demonstrate proof-of-concept to users.

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.

The user interface is particularly appropriate for prototyping. For example, rapid "storyboard" prototyping provides a software snapshot of a display screen that mimics what a user would see with the system. A prototype can be completed in weeks as opposed to months of laborious specification. An effective prototype can also enable users to sign-off on the new user interface and operational usage scenarios before hard system and software implementation decisions are made.

2.4 Identify Migration Piloting Needs

Because of the large number of factors in the legacy environment and the number of users who are potentially affected, pilot implementations on a small scale may be recommended to provide help in the migration effort. This activity should address the extent to which migration considerations suggest that pilot solutions are needed to validate system integrity, performance, and acceptance by users.

Besides their value in testing a technical solution, pilots also serve as a mechanism for providing informed user input and for achieving user buy-in. After determining the scope of the pilot, it is necessary to establish the degree of user involvement in validation and verification.

The Second DoD Product Line Workshop identified the following situations where a pilot approach might be appropriate [Bergey 99c]:

- There is a problem but it is unclear that consolidating legacy systems and migrating to a new system is a good idea.

- There is insufficient buy-in to the concept and user skepticism that migrating to another system will be able to meet their needs.

- The organization lacks a cooperative culture.

- There are limited resources to undertake a large-scale migration effort.

- The feasibility of migrating users from any of the existing legacy systems to a single new system has not yet been proven.

- There is currently too much instability operating and improving legacy systems to consider migrating.

- The organization is large and the pilot effort will be easily staffed.

To succeed, the pilot

- should not be on a critical path

- should be important to the mission or business

- should not require people who are already schedule constrained

- should not be tightly scoped and constrained

- should have a pre-existing advocacy group

Once the pilot is performed, it is important to evaluate its results and determine if the vision and goals are still practical or whether they should be scaled down, revamped, or possibly even abandoned. These kinds of decision points should be built into the migration plan.

2.5 Describe Roll-Out for Migration

The rollout of new capabilities needs to be carefully planned. Rollout tasks include

- describing phasing system builds

- determining support needs

2.5.1 Describe Phasing of System Builds

As part of developing the migration plan, an evaluation should be made of the number, type, and phasing of system builds that would best support customer priorities and user needs.

The plan should address the following set of questions:

1. To what degree should the system be rolled out and deployed in manageable increments to better accommodate customer priorities and the needs of the user community? What is needed for each increment from a customer and user standpoint?

2. What will be the impact and risk of rolling out the system's capabilities in a specific order and timeframe?

3. When and in what order can legacy systems be retired?

Milestones need to be identified in the migration plan that customers, developers, and users can all accept. A risk assessment should be performed before finalizing system build rollout plans to ensure planning estimates are reasonable, the approach is well conceived and in line with organizational priorities, and the potential impact on customers and users is acceptable.

2.5.2 Determine Support Needs

A level of support must be provided to assist users in using each build and transitioning to the new system. In addition, users often require operational assistance to help them understand the target system's capabilities and operations. Issues that need to be addressed include

1. What training and operational assistance will the users require?

2. What is the general level of migration support that will be required by the user community to ensure a successful migration?

3. How will user "buy-in" and target system acceptance be achieved?

4. How will unavoidable changes in legacy systems/products features and services be communicated to the user community?

The migration plan should focus on addressing these questions by targeting user support in the areas of

- training for users in learning and adapting to the new system's way of performing the same types of tasks

- a help desk and on-call technical assistance

- interim user manuals

- a trouble-reporting system

- technical assistance to users migrating to the new system

- opportunities for trial use of the new system

- system demonstrations to show existing legacy system users how the new system operates and how they can perform comparable tasks

- mentoring services to help key users transition their work to the new system

- workarounds to offset loss of support in critical areas or to overcome operational problems

2.6 Describe 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.

- Form a migration user's group for each major legacy system to address any lingering migration issues and problems.

3 Guidelines

Previous work has identified checklists for system evolution [Bergey 98], enumerated reasons why reengineering projects fail [Bergey 99b], and identified guidelines for migration efforts [Bergey 99a]. We derived guidelines from these reports and, in this section, apply them to the migration planning process.

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.

6. Avoid a "big-bang" approach to migration. Break the problem into "doable chunks" that correspond to the planned rollout of new system solutions:

   - Make certain the migration effort is appropriately scaled and commensurate with the organization's resources, skills, and current workload.

   - Establish priorities for migrating users of the existing legacy systems.

   - Give consideration to any organizational infrastructure improvements that can accelerate the migration effort or may be needed to overcome impediments in the current working environment.

   - Establish criteria to evaluate the level of difficulty of transitioning the user community corresponding to each legacy system.

   - Consider addressing high-risk migration issues first since their solution may have the greatest impact on the development effort and may determine the feasibility of migrating users of every legacy system.

   - Identify meaningful and measurable milestones to track progress.

   - Survey the user community as early as possible to obtain their insight into what they believe are the greatest impediments to moving to a modernized system.

   - Ensure that the scope of migration planning includes deployment, transition to full operational use, and phase out of any affected legacy systems.

7. Actively manage and monitor the migration effort:

   - Establish a goal-driven measurement program based on stakeholder needs to obtain visibility into the migration effort.

   - Schedule periodic reviews commensurate with the risks involved, organizational policies and regulations, vested interests of key stakeholders, and assurances sought by the sponsor and project manager.

   - Establish a tracking system to manage progress, problems, issues, and other action items that pertain to migration planning and execution.

4   Conclusion

Despite its importance to success, the development of migration plans has 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 technical note outlines the activities involved in migration planning. It presents steps that an organization can take in developing a migration plan. The note also provides a set of guidelines for implementing a migration plan.

References

[Bergey 97]   Bergey, J. K.; Northrop, L. M.; & Smith, D. B. *Enterprise Framework for the Disciplined Evolution of Legacy Systems* (CMU/SEI-97-TR-007). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1997. Available WWW. URL: <http://www.sei.cmu.edu/publications/documents/97.reports/97tr007/97tr007abstract.html>.

[Bergey 98]   Bergey, John K. *System Evolution Checklists Based on an Enterprise Framework*. Pittsburgh, PA: Carnegie Mellon University, Software Engineering Institute. Available WWW. URL: <http://www.sei.cmu.edu/reengineering/pubs/white-papers/Berg98/Checklist.pdf> (February 1998).

[Bergey 99a]   Bergey, J. K.; Smith, Dennis B.; & Weiderman, N. H. *DoD Legacy System Migration Guidelines* (CMU/SEI-99-TN-013, ADA370621). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, September 1999. Available WWW. URL: <http://www.sei.cmu.edu/publications/documents/99.reports/99tn013/99tn013abstract.html>.

[Bergey 99b]   Bergey, J. K.; Smith, D. B.; Tilley, S. R; Weiderman, Nelson H.; & Woods, Steven G. *Why Reengineering Projects Fail* (CMU/SEI-99-TR-010, ADA 362725). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, April 1999. Available WWW. URL: <http://www.sei.cmu.edu/publications/documents/99.reports/99tr010/99tr010abstract.html>.

[Bergey 99c]   Bergey, J. et al. *Second DoD Product Line Practice Workshop Report* (CMU/SEI-99-TR-015, ADA375845). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, October 1999. Available WWW. URL: <http://www.sei.cmu.edu/publications/ documents/99.reports/99tr015/99tr015abstract.html>.

[Feiler 93]   Feiler, P. *Reengineering: An Engineering Problem* (CMU/SEI-93-SR-005, ADA267117). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, July 1993. Available WWW. URL: <http://www.sei.cmu.edu/publications/documents/ 93.reports/93.sr.005.html>.

[Jacobson 99]   Jacobson, Ivar; Booch, Grady; & Rumbaugh, James. *The Unified Software Development Process*. New York: Addison-Wesley, 1999.

[Hofmeister 99]   Hofmeister, C.; Nord, R.; & Soni D. *Applied Software Architecture*. New York: Addison-Wesley, 1999.

[Royce 98]   Royce, Walker. *Software Project Management: A Unified Framework*. New York: Addison-Wesley, 1998.

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE August 2001		3. REPORT TYPE AND DATES COVERED Final
4. TITLE AND SUBTITLE DoD Software Migration Planning			5. FUNDING NUMBERS F19628-00-C-0003
6. AUTHOR(S) John Bergey, Liam O'Brien, Dennis Smith			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213			8. PERFORMING ORGANIZATION REPORT NUMBER CMU/SEI-2001-TN-012
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) HQ ESC/XPK 5 Eglin Street Hanscom AFB, MA 01731-2116			10. SPONSORING/MONITORING AGENCY REPORT NUMBER
11. SUPPLEMENTARY NOTES			
12A DISTRIBUTION/AVAILABILITY STATEMENT Unclassified/Unlimited, DTIC, NTIS			12B DISTRIBUTION CODE

13. ABSTRACT (MAXIMUM 200 WORDS)

The DoD often faces the challenge of migrating from legacy systems to new target systems. Such migration efforts represent a complex engineering problem. These efforts call for a migration plan to supplement the development plan. The migration plan addresses issues associated with phasing out legacy systems and moving to the new system. These issues include user interface compatibility, database compatibility, transition support, system interface compatibility, and training. By producing and implementing a migration plan, a development organization can help a user community make the transition in an orderly fashion.

This note describes migration planning, identifies influencing factors, outlines a set of migration planning activities, and offers a set of guidelines for the migration planning process.

14. SUBJECT TERMS Migration planning, software evolution, software reengineering			15. NUMBER OF PAGES 21
16. PRICE CODE			

17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL