



Enterprise Evolution: Modernization

The Transformation Process

Enterprise Evolution, a discipline in the legacy modernization domain, is a collection of tool-enabled disciplines that facilitates the understanding, improvement, migration, reuse and/or transformation of existing software systems. This is the third in a 3-part series of white papers that discusses the modernization options in an Enterprise Evolution modernization initiative, which may be applied to a variety of IT project initiatives.

While the first two white papers in the Enterprise Evolution series discussed the assessment and remediation of existing systems, this paper discusses the third leg of modernization – transformation. While remediation can be thought of as “modernizing systems in place”, the concept of transformation goes further by transforming existing data and application architectures into target architectures and environments.¹

Transformation includes moving and/or migrating applications based on heritage technologies into contemporary technologies, platforms and languages. In addition, transformation includes the capture, reuse and redeployment of existing data and business rules in object based languages, such as Java or C#, new data models, services oriented architectures and model driven architectures. This paper discusses the benefits of transformation as well as related disciplines that organizations can employ to derive the most value from existing software assets over the long-term.

Transformation Benefits

Transformation offers alternatives to development and package deployment projects and also provides ways to augment replacement initiatives through the reuse and migration of existing data and application architectures. Transformation:

- Streamlines the time and costs associated with eliminating or migrating obsolete technologies and platforms.
- Lowers the risks associated with deploying replacement systems based on the reuse of proven functionality extracted from existing systems.
- Reduces the costs and timeframes of migrating data into target data architectures, relational databases and package environments while improving the integrity of the end result.
- Supports business unit consolidation, mergers and acquisition initiatives through data and application architecture consolidation and integration.
- Enables the redeployment of proven, stable functionality in SOA, model driven and other agile architectures.

¹ The Object Management Group’s Modernization Task Force uses the term “Transformation” to discuss the shift to target architectures. eCube uses the term modernization interchangeably with transformation.



Language & Platform Migration

Moving from older, procedural languages, such as COBOL, C, FORTRAN, to object based languages, such as Java or C#, is a transformative process. This means that design and implementation paradigms used to create and deploy these object based languages differ from those used to create and deploy procedural languages.

These projects may include, for example, mainframe to open systems migration or a mid-range platform to J2EE or .NET migration. Regardless of the existing and target environments, these projects have several aspects in common. They include some combination of language transformation, data structure redefinition and mapping, data conversion, environmental artifact migration and redeployment to target platforms.

A number of modernization tools and techniques support language and platform transformation projects. In many circumstances, remediation activities can be used in combination with platform and language transformation projects to create more streamlined, more adaptable target architectures. User integration and middleware tools may also be used in conjunction with language and platform transformation projects to augment the final results.

Data Architecture Transformation

Data architecture migration continues to be a significant challenge for organizations attempting to align systems with business requirements. Users frequently claim that they cannot get the information they need, when they need and in the form that they need it in. Language migrations, middleware implementations, data warehouse deployments and other projects may only make a small dent in these information requirements.

Transformation of back-end, persistent data structures can address underlying limitations and misalignment between what users require and what IT can effectively deliver. This includes extracting and rationalizing data as defined in applications; reconciling system data usage with business definitions; developing a consolidated, normalized data model from these rationalized data definitions; and migrating data into target architectures.

The above data architecture transformation process would be phased in through a series of evolutionary modernization steps. Note that rationalization of data definitions refers to work performed during prior remediation stages of a project. Transforming data architectures can serve as a foundational step for application transformation projects or can stand along, based on one's goals.

Business Rule Analysis, Reuse & Redeployment

In addition to capturing, reconciling, redesigning and redeploying data under target architectures, organizations can also capture and reuse business rules from existing systems. The value of business rule reuse includes a reduction in the time and cost of re-specifying business logic from scratch; a reduction in the risks association with errant specification of functionality in target systems; and the reconciliation of inconsistent functionality across business areas.

It is important to note that a high-level, functional decomposition is highly recommended if business rule analysis is to be employed across multiple systems or a large sampling of application source code.



Further, performing data definition analysis as a prerequisite to business rule analysis creates essential constraints on logic search engines so that analysts can target appropriate subsets of business logic.

This tool-enabled, analyst-assisted process involves capturing, filtering, packaging, analyzing and redeploying business logic. Once program logic is extracted, it must be filtered, packaged and analyzed by subject matter experts before it can be called a business rule. The key aspect of this process involves using tools to find and distill vast amounts of functionality so that subject matter expertise can be leveraged effectively.

Transformation Support for IT Project Initiatives

The role of transformation in IT project initiatives is to leverage existing data and application functionality in the design, creation and deployment of a target platform, technology or architecture. Common projects that benefit from transformation support are as follows.

- **Platform / Language Migrations:** Understanding and remediating systems prepares them for migration, while transformation supports the actual design and retooling necessary to redeploy an application to a target language or environment.
- **Greenfield Development:** Traditional approaches involve a blank slate approach to replacing enterprise systems. These projects, which utilize the latest methods, tools and languages, are avoided due to the complexity of replacing entrenched systems. Transformation, however, allows analysts to reconsider replacement opportunities by leveraging existing data and application assets.
- **Package Deployment:** Implementing packages, including ERP and other off-the-shelf software, requires mapping packages to existing data and applications; decommissioning selected functionality; integrating the package into the current environment; and rationalizing, reconciling and migrating existing data into the package's data architecture. Transformation can play a key role in this analysis and reuse process.
- **Model Driven Architecture Migration:** Creating replacement applications using model driven options such as UML can be expedited through the analysis and reuse of existing application data definitions and functionality. At a minimum, analysts can verify the integrity of new designs by comparing those designs to existing data usage and business rules.
- **SOA Migration:** Migrating to SOA requires having certain functionality that can be deployed as services. Business rule reuse expedites this capability. In addition, data architecture transformation facilitates the cross-functional deployment of common services across an enterprise.

Transformation is the third leg of modernization. If organizations have projects that need to go beyond basic remediation, transformation provides significant benefits to the long-term value of existing software assets.

Visit ecubesystems.com to download **Part 1 (Assessment)** and **Part 2 (Remediation)** of this paper

William Ulrich is President of Tactical Strategy Group, Inc. and a strategic IT planning consultant. He is Co-Chair of the OMG Architecture-Driven Modernization Task Force and his latest book is entitled "Legacy Systems: Transformation Strategies".