

Modernizing EDI: Best Practices for Project Planning and Migration

An EXTOL International White Paper

This is the second in a two-part series of white papers on the subject of Electronic Data Interchange (EDI) modernization. In Part 1, we looked at EDI modernization drivers, options, and critical success factors. In Part 2, we examine best practices for planning an EDI modernization project and migrating EDI systems to modern EDI solutions.

Introduction

In part 1 of this white paper series, we introduced the distinction between Classic EDI and Modern EDI. Classic EDI is characterized by the batched interchange of standard electronic documents and flat files between trading partners, frequently over a Value-Added Network (VAN). Modern EDI extends the Classic EDI model with any-to-any connectivity, end-to-end integration, near real-time response, support for multiple, non-standard document syntaxes, on-demand visibility, and other value-adding capabilities.

Modern EDI is the result of many years of changes in business-to-business (B2B) trading practices and technologies. B2B changes usually arise organically. A single influential business that mandates trading partner use of a new document standard or communication protocol frequently triggers broader adoption within or even between markets. As a result of many such adoption cycles, businesses are finding that the EDI solutions they put in place years ago are no longer capable of meeting their requirements for trading partner integration. Classic EDI remains essential, but it's no longer good enough, by itself.

Modern EDI isn't a new kind of technology or EDI product category. Rather, it's an integrated set of capabilities that can be delivered in different ways. You can acquire modern EDI capabilities using on-premise middleware, cloud services, Integration-as-a-Service, or a combination. And the benefits of doing so are compelling. Businesses that invest in modern EDI capabilities are more agile, more scalable and efficient, and easier to do business with.

It is possible to realize some modern EDI benefits without wholesale replacement of your current EDI system. But realizing the full benefit of Modern EDI ultimately requires migration of existing EDI infrastructure, implementation assets, and processes. Just as there is no one modern EDI solution approach, there is no singular prescription for EDI modernization project success. But there are proven best practices that can reduce risk and make it easier to achieve your EDI modernization goals.

	Classic EDI	Modern EDI
Endpoint Connectivity	Limited to external partner integration using VANs and asynchronous direct connections	"Any-to-any" integration between partners, apps, data, communication services, and cloud services
Automation Scope	Simple inbound and outbound processes	End-to-end combinations of partner-facing and internal business processes
Process Latency	Asynchronous, batch-oriented processes	Mix of asynchronous batch and synchronous, near real-time processes
Document Transformation	Standard EDI document syntax and semantics plus flat files	Standard EDI plus XML, spreadsheet, and flat file documents
Application and Data Integration	Coded integration with applications and data	Configured, plug-in integration, aided by intelligent automation
File Transfer and Governance	Ungoverned, un-auditable file transfer and access activities	Managed transfer and governance of files exchanged with external and internal endpoints
Visibility	IT-dependent, "forensic" management and reporting	Centralized visibility and management of all integration activities, with alerts, self-serve dashboards, and on-demand reporting

EDI Modernization Planning Phase

Figure 1 depicts the gross staging of an EDI modernization project, consisting of a one-time planning effort, followed by a series of implementation sprints aimed at migrating existing assets to a modern EDI solution.

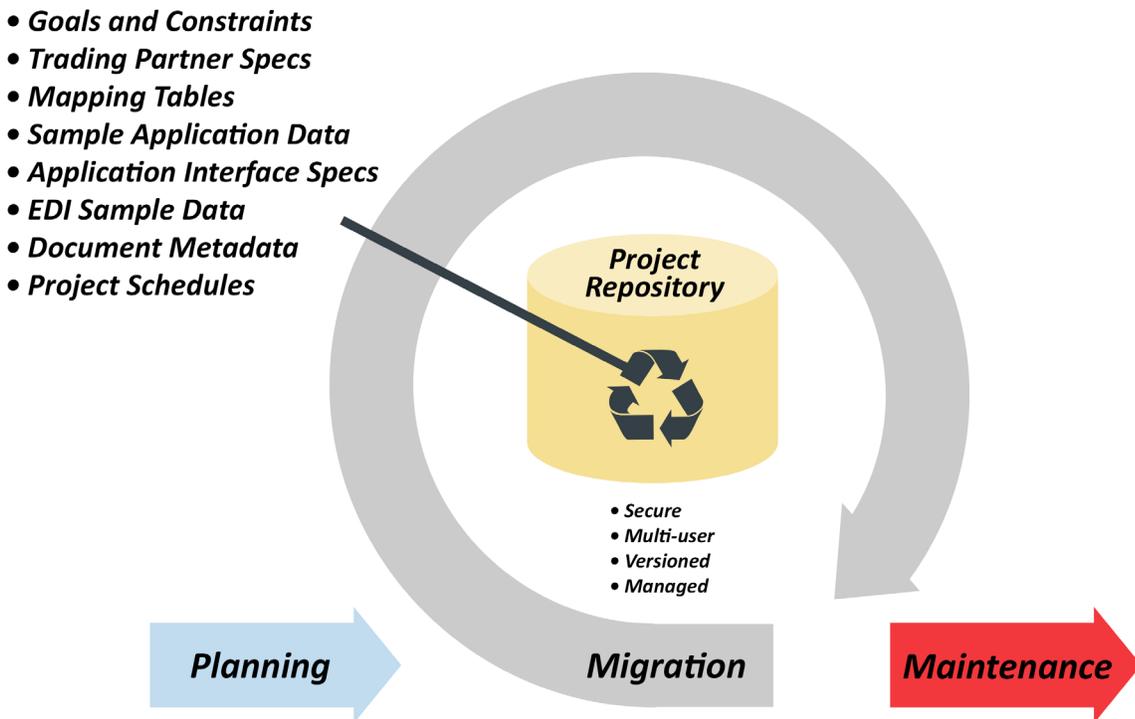


Figure 1: Sample planning deliverables for an EDI modernization project. Information captured during the planning phase provides a basis for modernizing EDI processes during the Migration phase.

The main goals for the Planning phase of an EDI modernization project are to:

- Identify unmet current needs and projected future needs
- Select an implementation approach and modernization solution
- Assign and train project staff
- Establish target system architecture
- Begin to populate a project repository with EDI specifications and legacy solution assets

The table below shows some of the main Planning phase tasks in a typical EDI modernization project. The actual set of tasks needed for a given project depend on goals, constraints, project scope, modernization strategy, and other situational factors.

#	Planning Tasks	Details and Options
1	Enumerate goals and constraints	New requirements, process improvements, delivery target, budget, change impacts, risks, etc.
2	Establish project infrastructure and policies	Agree on project structure, lifecycle processes, versioning and change management, set up project repository
3	Inventory legacy system	Catalog and populate project repository with trading partner specs, interface specs, sample EDI and application data, etc.
4	Perform gap analysis based on current and anticipated use cases	Middleware capabilities, integration processes and assets, tool functions, life cycle practices, exception-handling, reporting, etc.
5	Choose an implementation model	Self-implement, outsource, or hybrid. Consider strategic value, deadlines, budget, resource and skills availability, etc.
6	Identify supplemental resource needs	Needs for planning, implementation, design reviews, project oversight, etc.
7	Select modern EDI solution	Ensure fit to current and anticipated use cases, deployment preference, personnel needed, support requirements, etc.
8	Define roles and assign personnel	Assign coverage for all essential roles and tasks, whether outsourced or performed in-house
9	Train planning and migration staff	Ensure understanding of solution capabilities, life cycle processes, and best practices
10	Choose a modernization strategy	New development only, phased partner migration, phased integration layer replacement, en masse migration / implementation (for smaller cases)
11	Define integration architecture	Identify process, adapter, and mapping patterns, application interfaces, communications, exception handling strategy, performance considerations, etc.
12	Establish project schedule	Define scope, staging, and delivery estimates, partner implementation priorities, go-live policies, etc.

Best Practices for Planning

Use a Project Repository

Using a repository to capture planning documents and migration deliverables is a best practice, for several well-acknowledged reasons:

- Repositories facilitate information sharing among multiple project participants and constituents. When all or part of the project is outsourced, the repository becomes a critically important coordination mechanism.
- Repositories provide versioning and change management services that enable governance of document modifications and access.
- Repositories foster continuity and consistency across project phases and activities. This is especially important for ongoing maintenance of EDI assets after the migration project concludes.

Assign Project Staff According to Clearly Defined Roles

IT organizations generally struggle to staff EDI modernization projects with the right people and skill sets, for multiple reasons, including:

- Budget constraints that limit hiring and/or outsourcing options
- Unfamiliarity with EDI projects in general and / or modernization strategy in particular
- Skill gaps and over-allocation of the most experienced and highly skilled staff
- Staff (especially senior staff) preferences for and objections to various project responsibilities

No project staffing situation is ideal. Some adjustment of plans to accommodate available personnel and skill sets is normal and expected. But fundamental staffing errors like overlooking critical skill gaps or leaving important project functions unassigned can put project outcomes at risk.

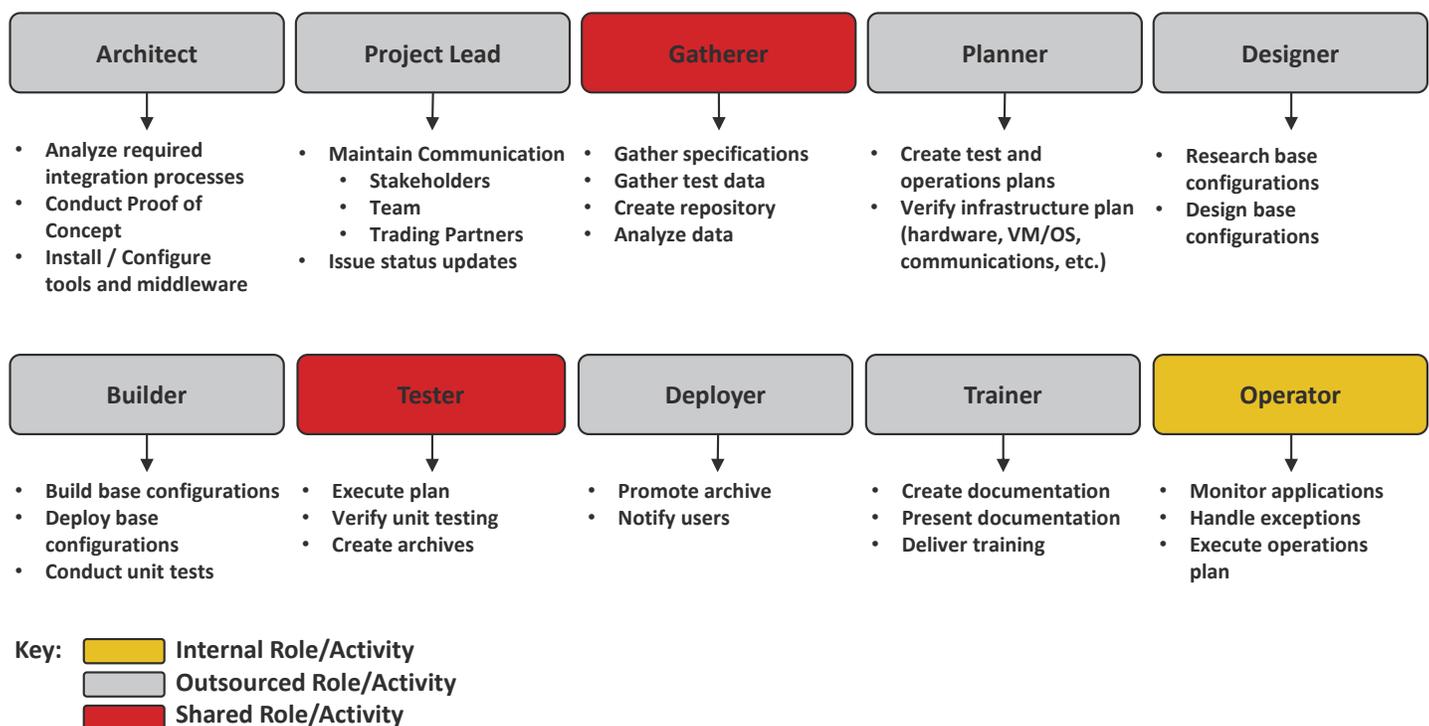


Figure 2: An example of roles and responsibility assignments for an outsourced EDI modernization project. Clear role definition facilitates work breakdown analysis, reveals staff supplementation needs, and provides a foundation for team member accountability.

A best practice to avoid such problems is to assign project staff according to clearly defined roles. Basing your staffing plan on a comprehensive list of project roles and responsibilities ensures that potential staffing issues are identified up-front, so they can be addressed before they put your project in jeopardy.

Some modernization project roles, like Project Lead, Business Analyst / Planner, and Builder, are almost always required, with Builders assigned later. Others, like Architect, Specifications Gatherer, and Trainer, can be important roles in larger projects, but assigned as minor, part-time duties, in smaller ones. Assigning more than one role to the same person is the norm, in most EDI modernization projects.

The precise set of roles and responsibilities needed for a given project depends on project scope, solution characteristics, methodology, and other factors. Ask your solution provider or (for outsourced projects) integration consultancy to recommend a project staffing model, then adjust to suit your project needs.

Conduct a Thorough Inventory of Your Existing EDI System

One of the first and most important tasks in a modernization project is to populate the project repository with an inventory of existing EDI system assets, including:

- Trading partner specifications (contacts, document standards, validation and mapping requirements, communication options and windows, security requirements, etc.)
- Sample data (EDI and application)
- Interface specifications (internal and external)
- Custom code extensions
- Change histories
- Open onboarding and change requests.

This best practice lays the foundation for detailed project planning and migration activities, by:

- Quantifying the number of partners, transaction types, and interfaces to be migrated
- Providing a basis for decisions about project structure and migration staging
- Identifying partner-specific customizations that will require individualized migration efforts
- Organizing baseline documents and assets for implementation sprints during the Migration Phase

Analyze Change Impacts Before You Choose a Replacement Solution

Replacing a legacy EDI solution affects the way you organize, implement, operate, and maintain EDI integrations. And modern EDI systems can be complex, exerting impacts in unexpected places. So before you choose a replacement solution, examine the differences between your current solution and candidate replacement solutions. Then consider potential change impacts at four levels:

- Middleware capabilities and services
- Partner onboarding and maintenance processes
- Runtime monitoring and management
- Quality of technical support and services

Use a life cycle activity model to analyze the impacts of different candidate solutions. Focus special attention on activities that are complex, time-consuming, or potential sources of errors or exceptions.

EDI Modernization Migration Phase

The main goals for the Migration phase of an EDI modernization project are to:

- Build templates for business processes, maps, adapters, and other integration deliverables to facilitate reuse
- Agree on priorities, processes, policies, and tempo for incremental migration
- Conduct implementation sprints to create, test, and deploy EDI integrations

The Migration phase of an EDI modernization project typically targets a combination of legacy migration and new integration objectives. New integrations that require modern EDI features (such as XML transformations or near real-time response) are frequently modernization drivers, and given top delivery priority. The subsequent migration of assets from the legacy EDI system is usually staged using a series of implementation sprints.

Migration of legacy EDI assets occurs using a combination of metadata-driven conversion, object reuse, and automation-assisted implementation. Direct, one-to-one asset migration is rarely feasible, because of integration model features that are not found in classic EDI systems, vendor-specific implementation differences, and intellectual property protections.

	Migration / Implementation Tasks	Details and Options
1	Develop templates and other reusable, shared objects	Create templates for processes, adapters, and base maps, sub-processes for notifications and other standard functions, etc.
2	Define scope and target for upcoming sprints	Identify implement / test / deploy targets for upcoming implementation sprints
3	Gather inputs to upcoming sprints	Populate repository with EDI implementation guides, SLAs, document definitions, sample data, interface specifications, test data, operations documentation, etc.
4	Perform migration / implementation sprint	Execute short “sprints” that implement and test a prescribed subset of integrations
5	Deploy and go live with new integrations	Test new integrations in parallel with existing system and deploy to production
6	Assess status and outcomes continuously	Continuously review implementation / migration activity during and after each sprint.
7	Refine plans, best practices as needed	Adjust migration tactics based on findings from continuous activity reviews
8	Report status to stakeholders	Keep internal sponsors, trading partners, and service providers apprised of status and changes
9	Conduct post-project review	Evaluate successes and shortcomings, update best practices, establish policies and processes for ongoing maintenance

The above table shows some of the main migration and implementation tasks in a typical EDI modernization project. Typical Migration phase tasks in an EDI modernization project. Migration and new implementation effort is decomposed into a series of sprints that deliver and deploy integration results rapidly and incrementally. Continuous assessment and plan refinement occur in parallel, along with pre-loading of the repository with inputs for the next sprint.

The actual set of tasks needed for a given project depend on goals, constraints, project scope, modernization strategy, and other situational factors.

Best Practices for Migration

Across modernization projects of all kinds and sizes, the same few migration activities consume the lion’s share of staff time. Data transformation mapping sits at the top of that list, followed distantly by document definition, endpoint configuration, adapter configuration, and process modeling.

In logistics, consumer packaged goods, and other businesses with a wide range of demand-side partner requirements, mapping and map testing can consume as much as 80% of project time. On the other hand, process modeling and adapter configuration consume comparatively little project time, because adapter and business process details do not, in general, vary across trading partners.

So the most effective way to shorten migration time – and a best practice – is to apply reuse and automation methods to the most time-consuming and complex migration activities. For example, cutting mapping time by 70% or more can reduce overall project time by 50% or more.

Figure 3 depicts where time is spent by activity in a typical small-to-medium EDI modernization project. Application of reuse and automation methods can reduce overall project time by as much as two-thirds. The next two sections below provide more information on how such savings can be achieved.

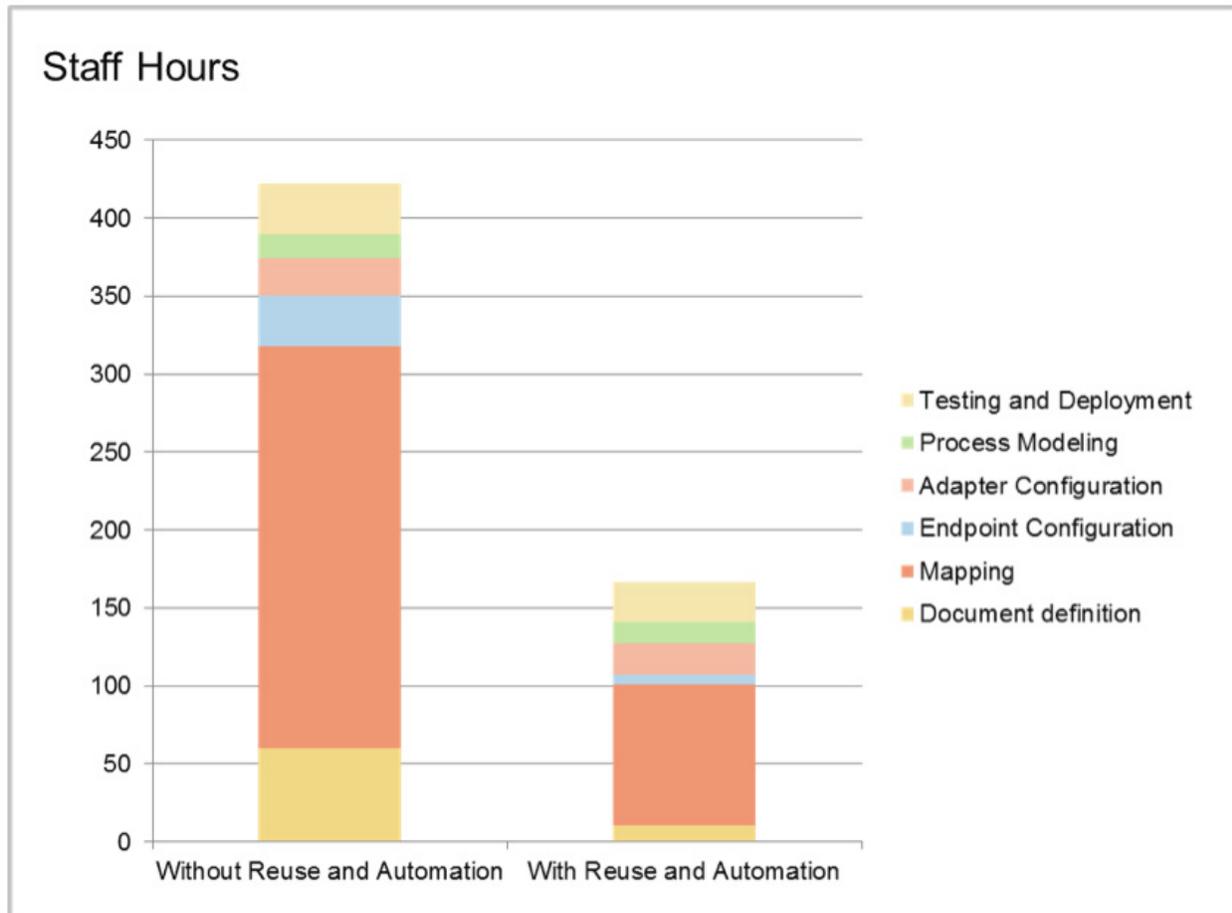


Figure 3: Complex, labor-intensive design-time activities, especially data transformation mapping, are prime targets for automation and reuse.

Analyze Integration Patterns and Promote Reuse with Templates

One important result from the Planning phase is an analysis of integration patterns. Integration patterns identify process, communications, interface, document type, and mapping commonalities across trading partner and EDI transaction cases. At the beginning of the Migration phase, that analysis guides the creation of reusable templates for business processes, maps, and adapters.

Building reusable templates is a Migration phase best practice, for several reasons:

- Template reuse cuts implementation time by avoiding the creation of assets that already exist.

- Templates promote consistency and enforce standards, improving overall system quality.
- Templates reduce testing time by propagating error corrections to multiple points of reuse.
- Templates reduce maintenance time by enabling changes to common behavior to occur in one place.

Figure 4 shows examples of reusable templates created from integration pattern analyses (left) and partner-specific deliverables created from metadata-driven tools and migrated EDI assets (right).

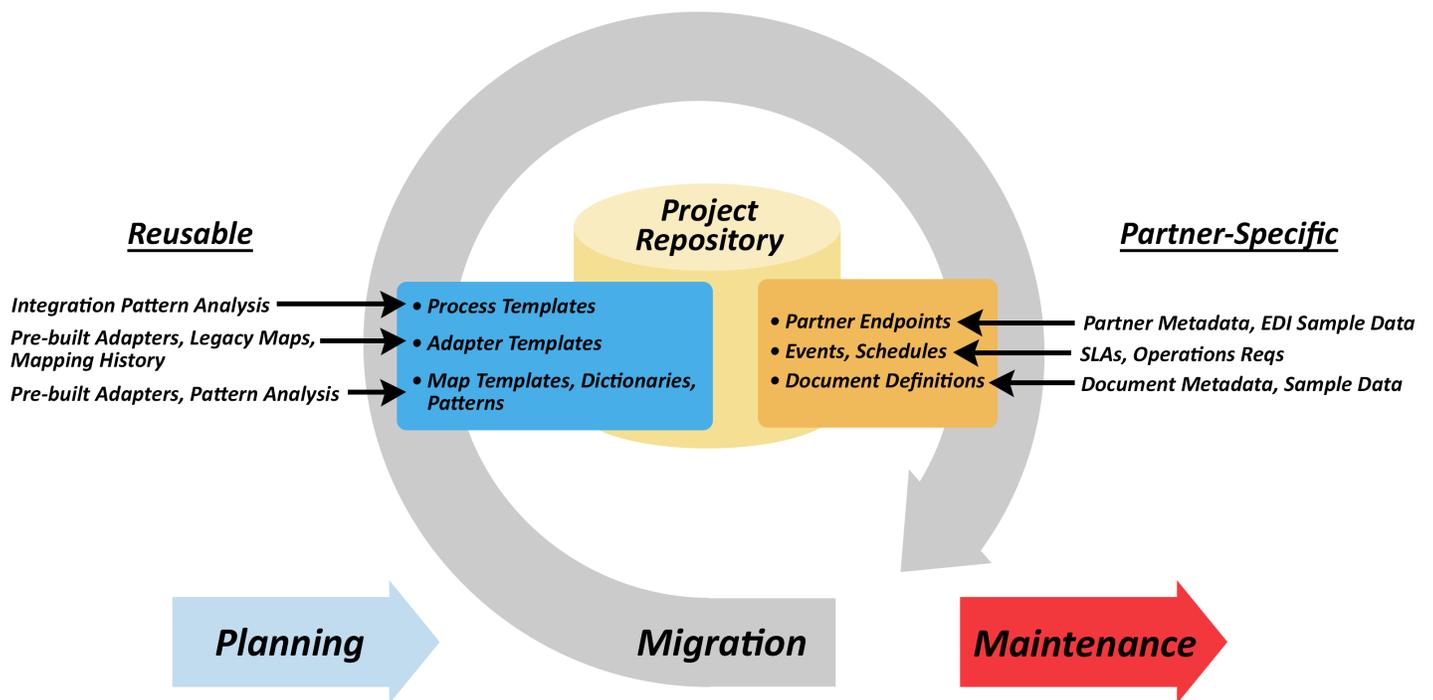


Figure 4: Sample inputs to implementation sprints in the Migration phase. Prior to each sprint, projects are populated with document schemas, digital certificates, and other partner-specific inputs. Together with process templates and other partner-independent inputs defined at the start of the Migration phase, this repository pre-loading practice maximizes sprint efficiency for faster delivery and deployment of integration results.

Employ Design-time Automation to Speed Partner-specific Customization

Design-time automation refers to the use of automated tools to generate or assist the creation of maps, business processes, and other integration assets. Automation can substantially reduce the time needed to create partner-specific assets, such as document schemas, maps, and events. In that way, it complements template reuse, which focuses on partner-independent integration assets.

Design-time automation is a Migration phase best practice for the following reasons:

- It reduces complexity by elevating human-tool interactions and hiding specification details.
- It speeds migration by exploiting external metadata, sample data, and other machine-readable inputs, in place of manual specification. It also automates operations on compound objects and collections, as when updating multiple maps that are affected by changes to a shared document.
- It improves overall system quality, consistency, and predictability by eliminating sources of human error.

Figure 5 depicts a typical sequence of activities that combine design-time automation with template reuse and manual configuration. The availability of design-time automation features varies widely between modern EDI solution providers. Look closely at mapping automation, document schema generation, and other capabilities and analyze their potential impacts on your migration schedule and subsequent onboarding requirements.

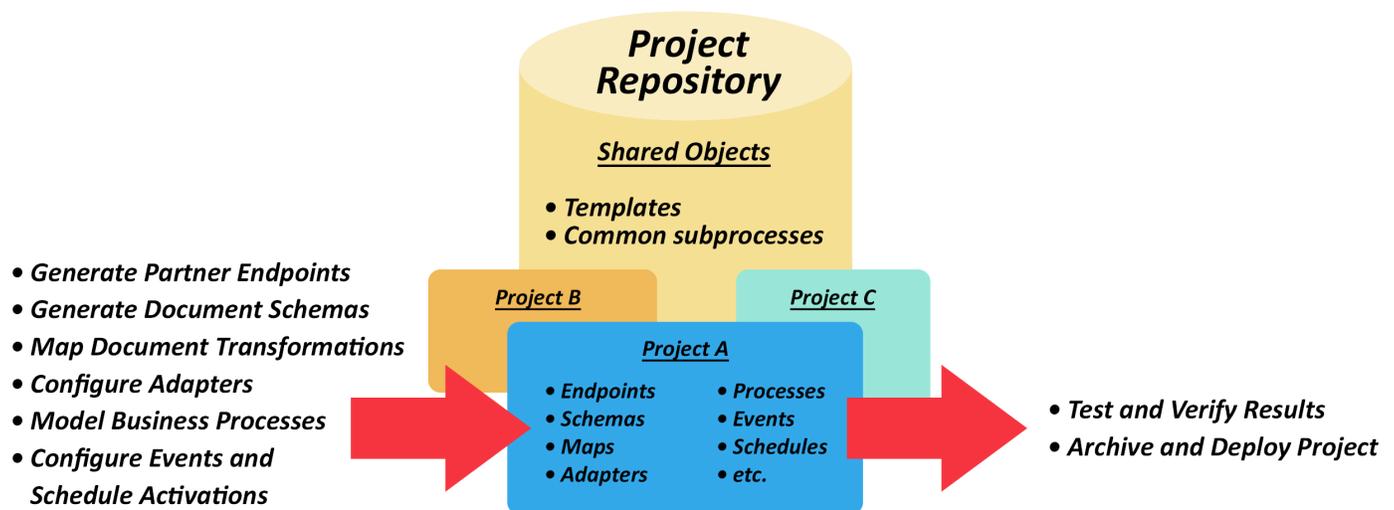


Figure 5: Sample activity sequence in an implementation sprint. Design-time tools combine templates and other reusable shared objects with maps, adapters, and other partner-specific deliverables to create new integration processes in the modern EDI environment. Tools employ automation where possible, generating objects from available metadata, sample data, and pattern histories.

Conclusion

The best practices cited in this paper have been culled from numerous EDI migration project experiences. But these recommendations are by no means exhaustive. Additional best practices include:

- Establishing formal Test and Production environment promotion policies
- Conducting a full life cycle dry run at the beginning of the Migration phase to identify and resolve process and policy issues
- Developing “triage” policies for handling various categories of runtime exceptions
- Designing integration processes with standard exception-handling and notification behavior
- Developing and documenting processes for routine backups and disaster recovery
- Employing vendor services as appropriate to shorten the learning curve, mitigate risk, and reduce time to value

Finally, some best practices have value even outside the context of EDI modernization. For example, maintaining a current inventory of EDI integrations, implementation assets, and documentation substantially reduces maintenance response time and facilitates ongoing reuse.

In the end, EDI modernization is a continuous concern, rather than a discrete event. Responding to new trading partner requirements is an ongoing requirement. But it’s also important to track unmet needs and anticipate future requirements and triggering events. Being prepared for an eventual transition to modern EDI is the best way to ensure a successful migration experience.

About the Author



Jim O'Leary, Vice President of Product Strategy at EXTOL International, is an expert on strategic business and IT project planning.

Jim has over 20 years of senior and executive management experience with software companies, and holds a deep understanding of current technological drivers across multiple industries.

e: joleary@extol.com

About EXTOL International

EXTOL International has been a provider of EDI integration software and managed service solutions for more than 20 years. The foundation of EXTOL's modern EDI offerings is EXTOL Business Integrator (EBI), a unified middleware platform and toolset for integrating trading partners, on-premise applications, cloud services, and enterprise data, in any combination. EBI is available as a software solution for self-managed EDI, and as an Integration-as-a-Service offering for companies that prefer to outsource EDI.

EBI automates customer and supplier interactions for manufacturing, retail, transportation, logistics, insurance, and many other industries. It comes complete with all of the capabilities you need for business-to-business (B2B) integration, including powerful process automation and data transformation; adapters for integrating partners, applications, and data resources; and a comprehensive, web-based dashboard for monitoring and analyzing integration activities.

EBI supports both classic and modern EDI requirements, including traditional batch and non-traditional real-time interactions. It includes X12 and EDIFACT EDI document standards, and supports integration of XML, flat file, and spreadsheet documents. The integrations you create with EBI are portable across IBM i, Windows, and Linux environments, as is the EBI product itself.

EBI integrates flexibly with business applications and data. It offers pre-built adapters for enterprise software products from Oracle, SAP, TMW Systems, and others, and supports both SOAP and REST services for integration with SaaS and other service-oriented applications. EBI can also integrate with internally-developed applications, using file system, database, web service, and other open interfaces.

EBI includes the EXTOL Integration Studio (EIS) design-time tool environment for creating and maintaining custom integrations that run on EBI. EIS includes advanced metadata-driven modeling and object generation tools that cut the time needed to create integrations and replace legacy integration assets. EIS also supports a very high level of object reuse, so you can bring new trading partners onboard faster, and with fewer errors.