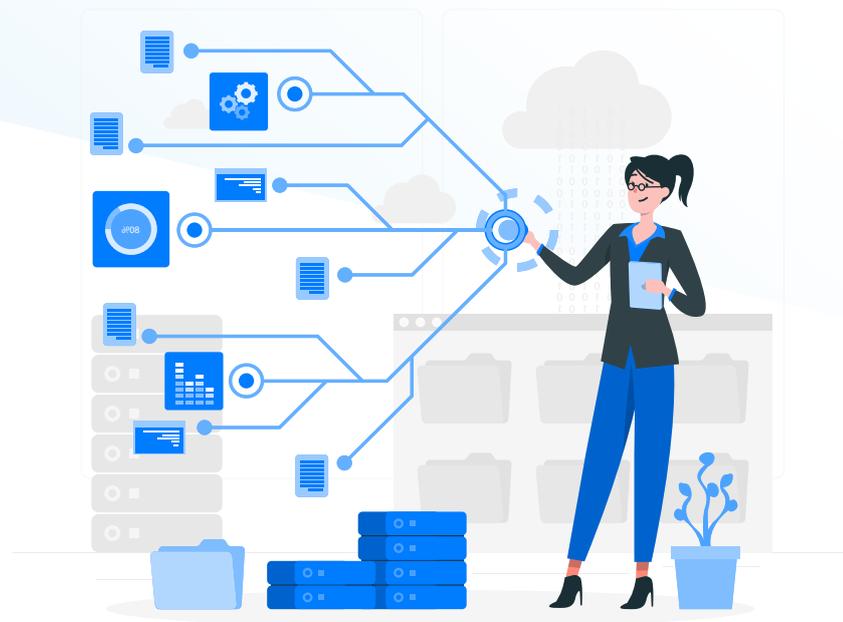


The Essential Guide to Application Data Migration



Importance of migrating active and historical data

Digital transformation demands the adoption of new tools and processes to support development teams' needs for agile, best-of-breed or modern solutions. Seamless data migration between legacy tools and the newly selected Commercial Off the Shelf (COTS) tools is key ensure a smooth and successful adoption of new tools and processes.

With a great deal of institutional knowledge and historical data locked in existing legacy tools, it is necessary to plan the migration strategy carefully. Preservation of cumulative knowledge with historical context and minimization of downtime are critical factors in the migration strategy. Poorly planned, sub-optimal or incomplete migration strategies can result in significant downtime, idle employees or loss of institutional knowledge and productivity. The loss of historical data may also have audit and compliance implications in addition to increased business risks. The ideal strategy minimizes downtime and maximizes the fidelity of migrated information, ensuring productivity, compliance, and business continuity.

True cost of data migration

Cost is an important consideration in data migration as the transition needs a significant investment in time and resources. Estimating the factors that add up to the actual cost of migration helps organizations plan better and allow for contingency in terms of system downtime or idle resources. However, it is not a straightforward formula because data migration impacts productivity and profitability in many ways. The following considerations help you quantify and measure the total cost of your migration and the savings/efficiency that you will make post data migration:

1. Data transfer costs

Whether you choose to outsource the migration project or do it in-house, there is a significant investment in building your migration competence so that your team can manage the project successfully. The larger the scope and duration of the project, the higher the cost.

2. Opportunity costs

Unavailability of systems translates into project downtime and loss of employee productivity that extends beyond the actual users of the affected system. A key factor to be considered is the business risks associated with extended downtime. For instance, the need for a security patch or update that is delayed due to unavailability of a system. Consider the overall impact of productivity loss, potential business risks and project interruptions or delays to evaluate the trade-offs if you are looking at a high downtime data transfer solution.

Systems downtime not only leads to productivity loss, it also increases the total costs as a result of pausing/restarting work.

3. Data corruption or information loss

For projects that include mission critical business information, there's an elevated risk of impacting operations in the event of data corruption or data loss. Unavailability or loss of access to information can have severe implications on profitability. Low fidelity migrations which often involve multiple manual steps can lead to data loss and data errors during migration. While they seem like the cheaper option, low fidelity migrations can have a long-term impact on team productivity and effectiveness. Typically, there is loss of versioning, history, artifacts, and context that leads to end-user frustration and performance challenges.

A full-fidelity approach ensures there is no data loss in the migration process even if there is a gap due to model mismatch between systems. This can be the deciding factor for businesses that work in highly regulated environments.

In addition to the cost of the migration project, there are other important considerations, including systems availability, security, fidelity, etc, which are often critical in selecting the migration strategy. For example, manual approaches have high security risks as sensitive data can be exposed during the migration. An automated factory approach is typically significantly more secure and reliable than approaches that rely on manual, human interaction activities.

Key factors for selecting the migration approach



Systems downtime or availability:

Factor in the impact of unavailable systems on your business continuity, security and idle users/system to decide the appropriate migration approach.



Data complexity:

The scale of your data, complexity, number of users and the need to customize fields and unusual data shapes will impact the migration choices and strategy.



Data quality:

Poor migration strategy results in data loss or corruption, rendering the data unusable in the target system.



Audits and Compliance:

Compliance issues can manifest while migrating content and the overall strategy must validate data in the context of audits and compliance.



Replicability of migration:

Data migration approach and configuration should be easily repeatable and replicable to achieve scalability to a large number of projects.



Data fidelity:

Applications comprise rich contextual information such as history, artefacts, embedded images, relationships, comments etc., raising multiple choices in terms of fidelity – full fidelity, high fidelity, medium fidelity and low fidelity.



Compatibility between source and target systems:

Disparity between the schemas and application structure of source and target systems requires additional effort on proper handling and modeling of the data before migration.



Security:

Data migration projects are susceptible to temporary lapses in privacy and security, often compromising sensitive information and customer data. Design a strategy with security issues in mind.



Training large numbers of users in a new system:

When migrating large numbers of users for a new tool, training and onboarding users' needs to be planned out in a timely, systematic way. Migrate the systems in prioritized tranches instead of a big-bang lift and shift.



Transformation (cleanse, merge, enrich):

Migrating often requires conversion of data from one format to another to resolve the model mismatch between two systems

Once you have taken these factors into account, identify the type of migration that works best for your business needs.

Identifying the migration type

1. Migrating within the same tool ecosystem

Organizations often opt to move from data center or on-premise servers to the cloud in order to lower operating expenses and increase agility, flexibility, and scalability. For instance, an enterprise may want to migrate data from its Azure DevOps Server or older Team Foundation Server (TFS) version, an on-site instance, to a common cloud-based Azure DevOps Services instance to improve collaboration in distributed teams and accelerate time-to-value by moving to cloud.

2. Legacy migration

Many legacy tools are outdated and difficult to maintain. Their siloed architectures do not integrate well with modern DevOps tools or project management tools. Migrating from a legacy ALM or issue tracking tool to a modern best-of-breed tool can lower the operational costs, empower teams with a feature-rich solution and real-time access to data and analytics.

3. Migration to support business re-organization

Business reasons such as restructuring, divestment, mergers and acquisitions or other such organizational activities often compel teams to migrate projects between heterogeneous tools where there is often disparity between the two systems. These types of migrations often require expertise to bridge the model mismatch between source and target systems.

4. Consolidation

Consolidation is required when multiple instances of an application are merged in a single instance – for example, three instances of Jira merged into one single instance of Jira. Consolidation can also be implemented across heterogeneous tools, such as merging two instances of X tool into a single instance of Y tool. There could be varied reasons for merging data, including better visibility, traceability, and transparency.

5. Splitting

Often data needs to be split among two or more instances or teams. This is typically the case with the divestment of a business unit within a larger company.

Selecting the right data migration strategy

Depending on the migration type, scale, and complexity of your project, you need to select the most suitable migration approach whether it is manual or automated. Manual or script-based approaches are time consuming, error-prone, and difficult to scale. Here are some of the use cases that support each approach:

1. Do-It-Yourself (DIY) script-based migration

It is possible to execute the migration project yourself with the help of scripts and REST APIs. However, with anything created in-house without relevant expertise or experience, there is always a risk of data inconsistency, loss of information and failure recovery. While many organizations have the raw technical capability to perform script-based migration, one must consider the time and energy that goes into the research, planning and execution and weigh it against lost opportunity and more strategic productivity.

This is a disruptive strategy and involves a fair amount of system downtime and loss of productivity. It also requires additional effort to map the entities and data, planning backward compatibility and configuration to migrate data between applications. There is limited scope to transform data or preserve the history and traceability of all migrated data.

2. One-time Data migration

One-time data migration is built on export-import framework with the purpose to execute one-time data migration from one system to another. Typically, these tools simply copy the data and there is limited support for data transformation during migration. While these migration tools support rich data fidelity, they require system downtime. Re-migration is required every time, any data issue is found in the target system during validation phase. Such migration tools are suitable for small scale migrations with less than 25 projects.

3. Enterprise-grade migration tool

An enterprise-grade migration tool is purpose-built for the various intricacies and challenges of large-scale data migration projects. It meets some of the most important criteria for fast, accurate and full-fidelity migration including:

- Zero downtime or non-disruptive data migration
- UI-based mapping that requires no scripting knowledge
- Factory approach for unmatched scalability
- Data migration with full context and history preservation
- Failure recovery or reconciliation mechanism to prevent data loss

Developing your migration plan

1. Identifying and defining the scope

Before migration, it is crucial to explore and assess the source, carry out a thorough audit and take complete inventory of all the assets that need to be migrated. Additionally, Source data issues or errors must be resolved before migration. For example, does the data within the source system fit the target application or does the data need further clean-up or transformation to be relevant and useful in the new system? Skipping source and target audits could lead to critical flaws in data mapping and a waste of time and resources.

2. Identifying and defining the suitable options for migrations

This phase involves drawing out the technical details of the solution and documenting the migration process. Identify the best option and approach for your migration and define the project timelines and concerns. Consult internal technical experts and get sign-off from stakeholders. Create a complete inventory of data/assets/linkages to be migrated including historical information and dependencies. Also, consider security plans for data if the data needs to be protected throughout the process.

3. Budgeting or determining access to funding

Migration can be expensive. Budget approval or finding access for funding a migration needs to be supported with the appropriate benefits and value of the migration to the organization. The ROI of a migration done right can outweigh the one-time cost of carrying out this transition.

4. Building or buying the migration solution

For smaller and simpler application migration with low complexity, a do it yourself (DIY) migration project might be feasible. But medium to large scale migration projects require expertise and experience. If you do not have the resources within your organization to manage the migration, it might also be worth consulting with a migration expert to evaluate various options and approaches.

5. Preparing teams for onboarding and training for new tools and processes

Adopting new tools and processes is challenging for teams who are being phased out of the old system. Onboarding the teams to the new application or platform requires a training plan and ongoing support to ensure optimal use of the new tool. There are several tools, consultants and training platforms that facilitate onboarding of team members. Also, factor in additional time and support required from service desk teams to resolve queries and challenges faced in user onboarding. Configure workflow, permissions and access for team members before they transition into the new system.

6. Execution

Execution must be planned considering the downtime if required. Before flipping the switch, select the right projects, specifying the data types to be migrated, providing user mapping details and checking source and target templates if customization is involved.

7. Validation

Migration testing and validation help ensure the accuracy of implementation and validity of data in the target system. After testing and go-live, setting up a process or system to audit migrated data and assets to verify the accuracy of the migration is equally important.

8. Reconciliation or error recovery

Regardless of the careful planning and execution, there are chances of failure within a data migration project resulting from project complexity, data consistency or heterogeneity of the systems involved. Your migration strategy needs a robust plan for failure and migration reconciliation or error recovery if things go wrong.

Final thoughts

Digital transformation initiatives rely on agility, adoption of modern technologies, and cloud-based applications. Application migration within the ALM ecosystem empowers DevOps initiatives and helps teams adopt best-of-breed tools without losing the historical data and linkages. The right migration approach should be tailored to an organization's migration requirements and use cases. At the same time, you don't have to compromise on data quality or productivity when you undertake a migration project. It is possible to retain accuracy, enrich and/or customize data for your new systems and perform a well-planned migration with zero downtime, and on budget.

Evaluate the ROI of different migration approaches, also considering factors such as downtime, impact of employee downtime/systems, compliance, delays in product release or the cost of inaccurate data/lost data.



1000 Elwell Ct, #101, Palo Alto,
CA 94303

Web - www.opshub.com

Email - sales@opshub.com

Phone - +1.650.701.1800