

The Hidden Challenge: Data Conversion in Software Implementation Projects



1. Introduction

Data conversion is often one of the most underestimated components of software implementation projects. Although it may appear straightforward on the surface, converting data from one system to another can be very complex and time consuming. This process is essential for ensuring that new systems function properly, and that valuable historical information is preserved. However, without proper planning and execution, data conversion can become a bottleneck that derails timelines and inflates project costs.

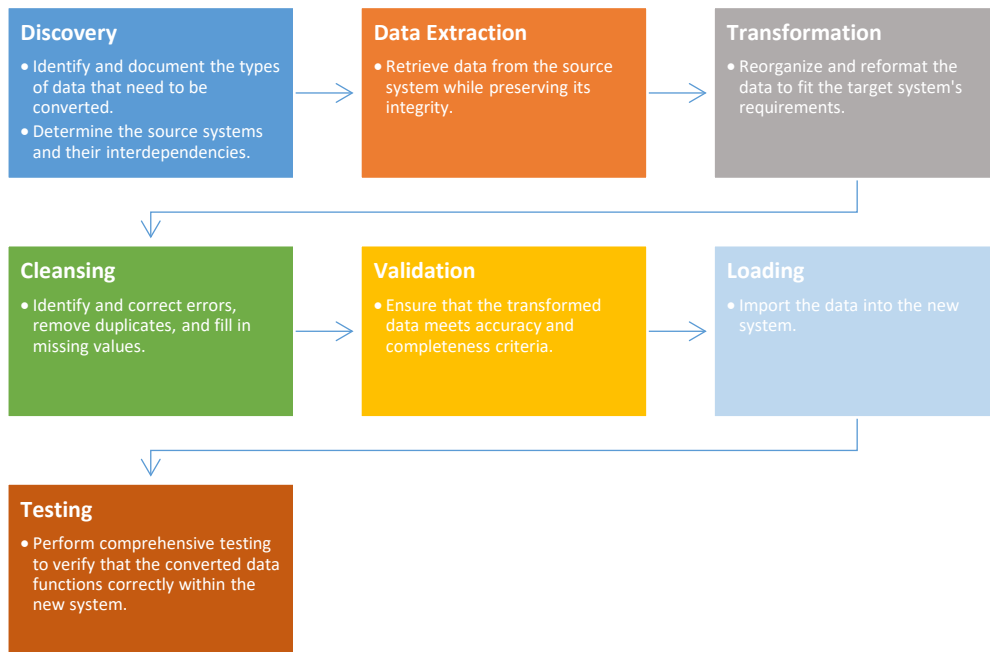
Many project planners fail to appreciate the complexities involved, assuming the process can be completed quickly without significant effort. This assumption is often incorrect because data conversion is inherently difficult. The challenges associated with data quality, differences in system architecture, and legacy technology can combine to create unforeseen obstacles.

This article explores the reasons why data conversion is often underestimated and outlines the steps typically involved in the process. It also delves into the specific factors that make data conversion difficult, examines the consequences of poor data conversion, and provides strategies to avoid common pitfalls. By addressing these challenges upfront, organizations can improve the likelihood of a successful software implementation.

2. Understanding Data Conversion

Data conversion involves transforming data from one format or system to another. This process ensures that data in a legacy system or other source is properly structured and compatible with the target system. It may include changing file formats, reconfiguring data structures, and cleansing data to align with new requirements.

The sub-phases of data conversion are:



3. When Is Data Conversion Needed?

You should plan for a data conversion effort anytime you are moving from one software solution to another. Examples include:

3.1 Legacy System Replacement

When outdated systems are replaced with modern solutions, data must be migrated to the new system.

3.2 System Migration

Transitioning from one software platform to another often requires reformatting and moving data.

3.3 Cloud Migrations

Migrating on-premises data to cloud-based solutions often involves significant effort.

3.4 Mergers and Acquisitions:

When organizations merge, their systems and data must be consolidated to ensure operational continuity.

4. Why Is Data Conversion Hard?

The complexities of data conversion often go unnoticed until the process is well underway. Many organizations begin with optimistic timelines and budgets, only to discover that they have underestimated the work involved. There are several reasons why data conversion can be a challenging endeavor.

4.1 Data Quality Issues

Inconsistent data, missing values, duplicates, or incorrect formatting in the source data can significantly complicate the conversion process. Poor data quality requires extensive cleaning and validation, which can add substantial time to the project.

Additionally, system usage can change over time. Users sometimes change what and where they store data in the system. They may do this to accommodate business process change or just convenience. Understanding the long-term system usage can help deal with this uncertainty.

4.2 System Differences

Different systems often have varying data structures, field names, and validation rules, making it difficult to map data directly between them. For example, imagine an organization migrating from an old CRM to a new one. In the old system, customer addresses might be stored as a single text field, while the new system separates them into street, city, state, and zip code fields. This requires custom transformation logic to split and map the data correctly, demonstrating how differences in system structure can demand significant analysis and rework to ensure compatibility.

4.3 Legacy Systems

Older systems may use outdated formats and technologies that complicate data extraction. Legacy systems may also lack documentation or support, making it difficult to understand how data is stored and structured. Poor or nonexistent documentation about the source data can lead to misinterpretations and errors during the mapping process. Without a clear understanding of the data's structure and purpose, inaccuracies can easily occur.

4.4 Data Dependencies

Complex relationships between data points in the source system can be challenging to replicate accurately in the new system. For example, consider a sales database where orders are linked to customers, products, and shipping details. If these links are not carefully maintained during the conversion, customer orders may be disassociated from their shipping information or product details, leading to data inaccuracies. Dependencies between datasets must be maintained.

4.5 Data Volume

Large datasets can take considerable time and resources to process during conversion. Processing millions of records requires robust hardware, software, and careful planning to avoid performance bottlenecks.

4.6 Time Constraints

Data conversion projects often operate under tight deadlines, adding pressure to complete tasks

quickly. This can lead to errors if corners are cut in the interest of saving time. This is a strong argument for doing your best to estimate this particular phase of the implementation.

4.7 Human Error

Mistakes in data mapping, validation, and testing can introduce inaccuracies into the converted data. These errors may go unnoticed until after the system goes live, leading to costly rework.

4.8 Scope Creep and Unrealistic Timelines

Changes to data structures or project goals late in the process can add unforeseen complexity. Expanding the scope of the project without adjusting timelines or budgets can cause significant delays.

5. Consequences

What happens when data conversion is underestimated? Your software implementation obviously runs increased budget and schedule risks but larger concerns around user adoption and regulatory compliance can also arise.

5.1 Delays in Project Timeline

Errors in the conversion process often result in extended testing periods and additional rework, causing delays to the overall project timeline. For example, issues discovered late in the process may require teams to backtrack and repeat multiple steps, including re-extracting and reloading data. This additional workload can extend the timeline by weeks or even months, forcing the project team to adjust milestones, delay system launches, and reallocate resources to mitigate the setbacks.

5.2 Increased Costs

Rework, additional staffing, and increased licensing costs can quickly escalate when data conversion issues arise. For example, hiring additional data specialists to resolve issues or purchasing extended software licenses can increase expenses beyond initial projections. These unexpected expenses can strain the project budget and require reallocation of resources from other parts of the project, ultimately affecting overall implementation timelines and outcomes.

5.3 User Adoption Issues

If end-users encounter missing or incorrect data, they may lose trust in the new system. This can cause resistance to adoption and diminished productivity. For instance, a sales team may find that critical customer information is incorrect or missing, forcing them to double-check records manually. This additional workload can create frustration, lower efficiency, and result in users reverting to old processes or workarounds. Over time, this lack of trust can hinder the system's full adoption and limit the expected return on investment.

5.4 Compliance and Legal Risks

Incomplete or inaccurate data migration can result in regulatory non-compliance, such as failing to adhere to data privacy regulations or industry-specific reporting standards. For example, if customer consent records are lost or incorrectly formatted during the migration, the organization may inadvertently violate privacy laws. This can expose the organization to legal penalties, fines, and reputational damage. Moreover, regulatory audits may become more difficult to pass, increasing the administrative burden on compliance teams.

6. Strategies to Avoid Underestimating

6.1 Estimate Conservatively

Account for unknown variables by estimating conservatively. An old project management joke advises taking the developer's initial estimate, adding a zero, and bumping the unit of time up by one. You may not need to go that far but examine estimates carefully. Remember that technical staff sometimes underestimate how long something is going to take. The reasons why they do this are many: they are trying to please their management, they underestimate complexity, or they overestimate their capacity.

6.2 Conduct a Comprehensive Data Audit

Assess the current state of data before the project begins. Identify inconsistencies, duplicates, and gaps to understand the full scope of work required. Data in systems, especially systems that have been around for a long time, can get weird, patchy, and inaccurate. Take the time to dig in.

“Data in systems, especially systems that have been around for a long time, can get weird, patchy, and inaccurate. Take the time to dig in.”

6.3 Involve Business Subject Matter Experts (SMEs) Early

Involve knowledgeable SMEs who understand the data and its history. They can provide critical context and insights. This can help identify potential issues, such as fields that have changed meaning over time, deprecated values, or undocumented changes. Additionally, SMEs can explain why certain workarounds were implemented in legacy systems and highlight hidden dependencies that could affect the data conversion process. Involving SMEs early ensures that nuanced data-related decisions are informed by domain expertise.

6.4 Pilot Conversions

If you can, conduct small-scale test migrations to uncover issues early and refine the process before full-scale implementation. The trick is to keep it small and focus on the data that is important.

6.5 Set Clear Ownership of Data

Assign responsibility for data quality and governance to specific individuals or teams to ensure accountability throughout the conversion process. This includes designating data stewards who understand the content and purpose of each dataset, as well as setting clear policies for data ownership and escalation in case of issues. These teams should also have the authority to enforce data quality standards and oversee regular audits to maintain consistency throughout the project.

6.6 Ensure Robust Validation and Testing

Implement comprehensive validation and testing processes at each stage of data conversion to catch errors before they become major issues. This is key and should be driven by the data quality and governance team in cooperation with the developers.

7. Conclusion

Data conversion is a critical but often underestimated component of software implementation projects. Many of the challenges stem from assumptions about data quality, system compatibility, and project timelines. Understanding these challenges and planning for them is essential to project success.

Project planners should prioritize data conversion as a core component of their implementation strategy. Allocating sufficient resources and time to this process can prevent costly delays and rework. Additionally, it is important for business stakeholders, technical leads, and data stewards to actively participate in the planning and execution phases. Business leaders should provide clear goals and support, technical teams must coordinate closely to ensure data integrity, and end-users should be involved in validation to ensure that the final data meets their needs. By engaging all key players in the process, organizations can foster accountability and collaboration, leading to a more successful implementation.

With proactive planning and attention to detail, organizations can overcome the complexities of data conversion and achieve a seamless transition to their new system. The benefits of a successful conversion include improved operational efficiency, data integrity, and user confidence.