

Ensuring Data Quality in Data Conversion

Most systems begin life the way scientists think the universe did—with a big bang. Conversion from an existing data source almost never goes smoothly and takes the better half of the new system implementation effort.

The Myth of Modern Systems

A common naïve belief holds that modern systems solve data quality problems. I remember well an HR executive confidently telling me over lunch that because his company was implementing a modern HR and payroll system, it would no longer have data quality problems. “PeopleSoft does not have data quality problems,” he said. How could I respond? I said, “I cannot tell you whether PeopleSoft has data quality problems, but your company certainly does, and they are not going away. In fact, they are about to get much bigger.” He did not believe me—I suppose because I had rained on his parade. A few years later, that company’s new HR executive called me for advice about their data quality strategy.

I still vividly remember one of my first major data conversion projects. I was on a team implementing a new pension administration system. We needed to convert employee compensation data from the legacy HR database. The old data was stored in much detail—by paycheck and compensation type. The new database simply needed aggregate monthly pensionable earnings. The mapping was trivial: take all records with relevant compensation types (provided as a list of valid codes), total the amounts for each calendar month, and place the result into the new bucket.

The Reality

The result was disastrous. Half of the sample records I looked at did not match the summary reports printed from the old system. A meeting was called for the next morning, and in the wee hours of that night, I had the presence of mind to stop looking for bugs in the code and to poke into the source data. The data certainly did not add up to what was showing on the summary reports, yet the reports were produced from these very data! This mathematical puzzle kept me up till dawn. By then, I had most of it figured out.

Our list was missing half a dozen compensation codes included in the aggregate amounts. In fact, compensation codes were even missing from the data dictionary. Certain codes were used in some years but not in others. Records with negative amounts—retroactive adjustments—were aggregated into the previous month, to which they technically belonged, rather than the month of the paycheck. Apparently, the old system had a ton of code that applied all these rules to calculate proper monthly pensionable earnings. The new system was certainly not programmed to do so, and nobody had remembered to indicate all this logic in the mapping document.

For eight weeks, we conducted extensive data profiling, analysis, and quality assessment in order to complete this portion of the project—an undertaking estimated and budgeted at one week. We were lucky, though; the problem was relatively easy to expose. In many projects, the data is converted based on mapping specifications that are ridiculously out of sync with reality. The result is predictable: mass extinction of the data and the project teams.

Hidden Hazards

What makes data conversion so hazardous? The greatest challenge in data conversion is that actual content and structure of the source data is rarely understood. More often, data transformation algorithms rely on the theoretical data definitions and data models, rather than on information about actual data content. Since these metadata are usually incomplete, outdated, and incorrect, the converted data looks nothing like what is expected. The solution is to precede conversion with extensive data profiling and analysis. In fact, data quality after conversion is directly (even exponentially) related to the amount of knowledge about actual data you possess. Lack of in-depth analysis will guarantee significant loss of data quality.

Another hazard is poor understanding that every system is made of three layers: database, business rules, and user interface. As a result, what users see is not what is actually stored in the database. This is especially true for legacy systems, which are notorious for elaborate hidden business rules. During the data conversion, the data structure is the center of attention. The data is mapped between old and new databases. However, since the business rules of the source and destination systems differ markedly, this approach inevitably fails. Even if the data is converted with utmost accuracy, the information that comes out of the new system will be totally incorrect.

So far I have talked about the data problems introduced by the conversion process; however, the source data itself can pose a hazard. Existing erroneous data tends to spread like a virus during conversion. Some bad records are dropped and not converted. Others are changed by the transformation routines. Such “mutated” errors are much more difficult to identify and correct after conversion. Even worse, bad records affect conversion of many correct data elements. A data cleansing initiative is typically necessary and must be performed before, rather than after, conversion.

Summary

The quality of the data after conversion is directly proportional to the amount of time spent to analyze, profile, and assess it before conversion. In an ideal data conversion project, 80 percent of the time is spent on data analysis and 20 percent on coding transformation algorithms. In practice, however, this rarely occurs. We are trapped by the infamous ETL—extract, transform, and load. Data conversion is somehow perceived as the process of moving data from system A to system B with some transformations along the way. This is so naïve and so wrong. It is like saying that moving to a new house is a process of loading the furniture on a truck, driving to the new place, and unloading the furniture in the new house. What you really need to do is analyze what you have, how (and if) it will fit in the new layout, throw away some old things, buy some new things, and only then rent a truck. The physical move is the easiest part if you did the analysis properly beforehand. The same is true for data conversions.

Ensuring data quality in data conversion is the most difficult part of any system implementation. It needs your team’s full attention.