# Chapter 34
# Building Data Warehouses Using Automation

**Nayem Rahman**
*Intel Corporation, USA*

**Dale Rutz**
*Intel Corporation, USA*

## ABSTRACT

*Software development is a complex endeavor. While significant benefits can be achieved, the process is often laborious, time consuming and error prone requiring multiple iterations in order to achieve the desired result. Issues arise for numerous reasons – coding defects, unclear requirements, migration challenges, lack of convention, and inadequate testing to name a few. When convention and automation are introduced into the software development lifecycle there are significantly fewer opportunities for failure. Automation also allows for shorter development windows. Generally there are fewer errors throughout testing, with the bulk of those being found in unit and functional testing, far before the users get involved in systems acceptance testing. A data warehouse consists of multiple subject areas in which many tasks are common and should be automated for the sake of efficiency and enforcing convention. This article discusses a set of tools that can be used to automate writing data warehouse objects. The article also provides statistics of time saved using automation.*

## 1. INTRODUCTION

Building a data warehouse is expensive because it is a labor-intensive and time-consuming endeavor. There are many activities involved with building a data warehouse, including requirements gathering and analysis, source data analysis, source-target data mapping, data transformation logic, extract-transform-load (ETL), design, and data warehouse loading (Wells, 2014). Issues with resource assignments are typically encountered throughout the development life cycle, especially when dealing with a waterfall development methodology. At some points in the project life time, the developers may be working night and day to meet the deadline (Rahman et al. 2011; Herschel 2012), while during other phases such as requirements gathering developers may be largely idle. Typically, at some point in the project's life cycle, it falls behind

DOI: 10.4018/978-1-5225-8060-7.ch034

schedule raising concerns of stakeholders. To resource manpower in lengthy projects managers have to make extraordinary efforts to deal with schedule pressures and late software projects (Williams et al., 2004). Often they respond by adding more man power to stay on schedule. This method of addressing schedule slippage is often counterproductive. Through his law, "adding manpower to a late software project makes it later,'' Brooks (1995) asserts that the assimilation, training, and intercommunication costs of adding new team members outweigh the associated team productivity gain in the short term.

Building data warehouses using automated tools can help to overcome manpower issues, schedule pressures, and late delivery of projects. Data warehouse automation (DWA) tools and technologies can accelerate warehouse development while simultaneously improving quality and consistency (Wells, 2014). A set of automated tools can help the extract-transform-load (ETL) programmer to perform ETL development more quickly and with higher quality. Large manual ETL efforts require significant resources. It is difficult to accurately estimate the manual effort involved especially when it spans multiple resources and therefore the work often takes longer than expected to complete. Organizations have been working to remove redundancy, step up automation efforts, and minimize project slippage by eliminating manual work during the project life cycle. It is critical to find ways to successfully implement software development projects on time, as these initiatives typically contribute directly to an organization's bottom line.

An automated ETL development approach in data warehousing projects is proving essential to the success of data warehousing projects (Brobst et al., 2008). Life cycle analysis (LCA) is useful to identify redundancy as well as scope of work needed for the manual efforts in each stage of a project, including schema design, ETL development, testing, and deployment into production. Use of automated tools helps to minimize development time and eliminating human error, which then reduces testing time. Data Warehouse Automation tools can automate data modeling, schema creation, data mapping, ETL coding, and batch cycle runs (Nicholson, 2014). Introducing various automated tools can help to reduce inefficiency, redundancy, and wasted time, while introducing conventions in coding, and reuse opportunities. The major driving force for shifting to an automated ETL-development-based approach is the result of data warehousing applications projects, which are continually put at risk for late project implementation and cost overruns. In data warehousing there is growing pressure to satisfy stakeholders' needs by using innovative development methodologies, delivering products flawlessly and minimizing product delivery time (Morien, 2005) and risk.

Building a data warehouse requires creation of numerous ETL, schema and validation objects in addition to integrating the various pieces of code. Projects need to create tables to store data, write stored procedures to do transformations and load target tables, create views for reporting tools and produce data validation scripts. As part of data warehouse effort database objects need to be synchronized in all data warehouse environments including development, testing, and production. An automated tool (Rahman et al. 2010) can be used to synchronize objects in different data warehousing environments. Lastly, batch cycles run in a data warehouse can be made metadata model-driven (Rahman et al. 2012).

## 2. LITERATURE REVIEW

Data warehousing is a critical layer in information technology (IT), as it lays the foundation for all data mining and business intelligence initiatives (Yermish et al. 2010). Data warehousing and data management have been identified as one of the six physical capability clusters of IT-infrastructure services required