The process of planning and executing SQL Server migrations can be complex and risk-prone. This is a case where the right approach and implementation of best practices can help you avoid the pitfalls of migration, minimize associated risk, and complete the migration more quickly.

This Dell EMC Perspective provides an overview of how we’ve applied the “lessons learned” over the course of engagements around the world to a set of best practices that help reduce risk and accelerate the benefits associated with SQL Server migration projects.
# TABLE OF CONTENTS

**CHALLENGES** ................................................................. 3

**GET OFF TO THE RIGHT START** ........................................ 3

**IDENTIFY & MAP INTERDEPENDENCIES** ............................ 3
Take third-party applications into account .................. 4
Address the legacy ......................................................... 4
Control your costs ........................................................... 4

**PROTECT YOUR DATA AND UPTIME** ................................. 5

**LOOK AT THE ENTIRE DATA PLATFORM** ............................ 5
Test and test again .......................................................... 5

**CONCLUSION** ................................................................ 5
CHALLENGES

Too often we think of SQL Server simply as a database engine, while it is actually an entire data platform supporting data integration, data warehousing, analytics, and reporting. There’s a great deal to consider when migrating from one version of SQL Server to another.

It is important to consider the broad range of SQL Server features that your organization may be leveraging. Migrating and upgrading these features can be significantly more complicated and time-consuming than just upgrading the databases alone.

Over the course of thousands of consulting engagements with customers around the world, we’ve learned a great deal about migrations and how best to successfully navigate the process. We’ve applied these “lessons learned” to a set of best practices that help reduce risks and accelerate the benefits associated with a SQL Server migration project.

GET OFF TO THE RIGHT START

You can’t manage what you can’t measure. You need to assess where you are in terms of current deployments, version levels, critical features, and performance levels. Discovery is the key to success.

Start with an inventory of all servers running SQL Server. The Microsoft Assessment and Planning (MAP) toolkit can help identify all instances of SQL Server deployed in the enterprise with specific versions of SQL Server running on each server. It is common to discover many instances of SQL Server that are not known to either the IT or the database administration (DBA) teams.

Understand all the SQL Server features that are currently implemented, including the business intelligence stack, high availability (HA), disaster recovery (DR), and others. Run performance analysis on all servers as early as possible in order to provide baseline performance characteristics for both databases and servers. This information will later be used for consolidation analysis and capacity planning—including memory, compute, network, and even storage platforms. CPUs are often underutilized on both physical and virtual SQL Server instances.

There are many industry tools available to help with the performance analysis, including native Windows and SQL Server tools. Microsoft SQL Server Upgrade Advisor, for example, can identify issues that may have an impact on upgrade to more recent versions of SQL Server.

IDENTIFY & MAP INTERDEPENDENCIES

Develop a comprehensive inventory of SQL Server database dependencies, such as OLTP. Migration and consolidation includes more than just databases. There are a number of dependencies such as SQL Server logins, SQL agent jobs, linked servers, and other configurations that must be part of the plan for migration and upgrade.

Typically linked servers are configured to enable you to use distributed queries in SQL Server database s or in other database products and query both as if they were in the same database. Third-party application dependency tools may also be deployed. That will require analysis and planning to ensure the appropriate tools are available in the target infrastructure.

Create and maintain a catalog of application-to-database dependencies prior to planning an upgrade. This is a time-consuming exercise that will require collaboration with many subject matter experts in the organization.
Plan time and resources to catalog and map applications to databases. Scripts can be used to determine the login name and machine name for connections being made to individual databases. However, identifying an application based on connection information may require the involvement of IT and/or line of business (LOB) stakeholders.

**Take third-party applications into account**

Evaluate supported versions of SQL Server for third-party applications—which may comprise myriad products that rely upon a SQL Server backend. Third-party applications may have a variety of requirements specific to a particular version of SQL Server, the infrastructure design, and the feature sets that are enabled. In many cases it may not be obvious that an application has a dependency on a SQL Server database infrastructure.

The upgrade path and supported SQL Server versions for LOB applications can vary from vendor to vendor and may require upgrading the application to a more recent version. That could involve additional licensing fees and vendor support costs.

Consider the broad range of SQL Server features that your enterprise may be leveraging. Migrating and upgrading these features can be significantly more complicated and time-consuming than just upgrading the databases alone.

**Address the legacy**

Take into account the fact that legacy custom data solutions may have been in place for several years. Subject matter experts with knowledge of the functional requirements and technical design may no longer exist within the organization.

Since your initial implementation of the solution, some vendors may have been acquired, be out of business, or simply no longer willing to support a legacy application. Consideration should then be given to trade-offs between the long-term value of supporting the existing application versus migrating to a new application.

In that case, you may need additional time to review the application design, to understand the business and functional requirements, to evaluate the technical design elements, and to identify interdependencies that may exist between the application and other components within the infrastructure.

Taking these first steps will show you how your organization can get from where it is today to where it needs to be. It is not just about the logistics of the move—you will also have the right answer when it comes to the complete infrastructure required to support the new environment.

**Control your costs**

Beyond creating an inventory of application dependencies on the data infrastructure and rating the applications for migration/upgrade criticality, we look at licensing, business, and support impact. Don’t forget to consider how you will maintain the stability of the business during the migration and mitigate risk throughout the process.

The costs of migration can increase quickly as you consider SQL Server licensing costs. Licensing is charged per CPU in SQL 2000, 2005 and 2008. The licensing model changes to per Core in SQL 2012 and later versions, which emphasizes the need to evaluate workloads at the CPU level.

Without proper planning, licensing costs could increase significantly. Remember that Enterprise Edition with Software Assurance can be licensed at the virtual machine (VM) host level, allowing for unlimited number of SQL Server VMs to be provisioned.
PROTECT YOUR DATA AND UPTIME

Plan to evaluate and standardize HA and DR solutions based on the target infrastructure and SQL Server version you intend to implement. Consider alternatives to Microsoft SQL Server Clustering such as SQL Server AlwaysOn Availability Groups in SQL Server 2012 and later versions, as well as VM host clustering and physical infrastructure DR solutions.

Many environments experience employee turnover, which can lead to the loss of code base tribal knowledge. An up-to-date source control repository will help address any questions that can surface during a SQL migration and upgrade.

Accounting for application outage windows and migration time frames will also be of critical importance. Migration time frames can vary based on application data storage requirements, HA and DR solutions, as well as application interdependencies. In many cases migration events will need to be planned based on such interdependencies to minimize risk and maximize utilization of test resources.

LOOK AT THE ENTIRE DATA PLATFORM

As we said at the outset, when you’re migrating to a more recent version of SQL Server, you’ll need to consider the entire data platform supporting data integration, data warehousing, analytics, and reporting.

There are three main components of SQL Server that are important for your BI solutions: SQL Server Integration Services (SSIS), SQL Server Analysis Services (SSAS) and SQL Server Reporting Services (SSRS).

It’s important that you don’t underestimate the time and effort to migrate and test BI solutions. Data warehousing, ETL, analytics and reporting can involve complex custom code with many dependencies outside the SQL Server platform. The upgrade path may be complicated if the move is from SQL 2000 or SQL 2005 to SQL 2012 and above.

Remember to plan for developer support to refactor business intelligence (BI) objects that are affected by the upgrade path. This is an excellent opportunity to confirm that all BI and custom SQL Server artifacts are properly updated in your source control repository. Understanding where the true “source of record” resides will greatly reduce migration risk.

Test and test again

Plan for application testing and leave enough time to adequately prepare and execute. Application experts and IT staff need to be prepared for testing applications—activities that may prove to be time-consuming. You’ll need to ensure that high-level test plans and test cases exist for business critical applications.

It’s only after you complete the assessment and planning activities that the design should be finalized and execution of the migration itself should take place. At Dell EMC, we work with our customers to design and deploy a modern SQL infrastructure with validation of processes, tools, and procedures to successfully complete the database migration. This leads to creation of the final migration map, future-state architecture, and the migration itself.

CONCLUSION

Dell EMC’s recommended approach to migrations is very straightforward. A two-phased plan beginning with efficient assessment, planning, and architectural requirements gathering is the right way to get started. In most cases the activity in that initial assessment phase will pay for itself in license and infrastructure optimization alone.
It’s only after you complete those essential planning activities that you can move with assurance to finalizing the design and feeling confident about the success of the migration itself.

The end goal is to deploy a modern, cost-efficient SQL infrastructure with validation of processes, tools, and procedures to successfully migrate the SQL Server data platform.

The process of planning and executing a SQL Server migration is often complex. This is a case where the right approach and implementation of best practices can help you avoid the pitfalls of migration, minimize associated risk, and complete the migration more quickly.