Redgate's Database Lifecycle Management (DLM) approaches

	State- or model-based approach	Migrations-based approach
What's the approach?	A single step between the current database state and the desired database state.	Individual changes are captured in multiple small scripts, which are applied in sequence to effect larger, iterative database migrations.
Key advantages	1. Database developers only concern themselves with defining the desired end state, not how the transition occurs. 2. Developers do not need to have strong database expertise, as the deployment script is auto-generated in seconds prior to upgrade using **SQL Compare** diffing technology. 3. For complex changes, custom migration scripts can be written in **SQL Source Control** to override default behavior. 4. **SQL Source Control** allows database changes to be committed to source control directly from SQL Server Management Studio. 5. **SQL Source Control** supports both dedicated and shared development database environments.	1. Database developers are responsible for writing "migration scripts" that become the building blocks for the eventual deployment script. 2. Deployment is repeatedly pre-validated during development, rather than fewer times later on in the project lifecycle. 3. There is no ambiguity in how an upgrade will be performed. 4. **ReadyRoll** migration scripts can use environment-specific variables, allowing server and database names to be different in each environment. 5. **ReadyRoll** projects can be part of a Visual Studio solution, and check-ins for application and database changes are treated the same.
What to buy	**SQL Toolbelt** Use SQL Source Control for a state-based approach.	**SQL Toolbelt** Use ReadyRoll for a migrations-based approach. As ReadyRoll isn't fully integrated into the SQL Toolbelt yet, please download it from redgate.com/ReadyRoll and contact sales for information on how to purchase it as part of the SQL Toolbelt.

redgate

How the SQL Toolbelt helps with DLM

	State- or model-based approach	Migrations-based approach
Development	Schema and data comparison and synchronization with **SQL Compare Pro** and **SQL Data Compare Pro** • Quick overview of differences between SQL Server databases • Accepting or rejecting database drift Increased team productivity with **SQL Prompt** – the leading tool for T-SQL code completion in SQL Server Management Studio (SSMS) and Visual Studio (VS) • Safe refactoring, code formatting, snippet creation and sharing, and more Additionally, teams benefit from realistic test data (**SQL Data Generator**), an integrated database unit test runner (**SQL Test**), quickly finding text fragments in SQL Server objects (**SQL Search**), automatically generating database documentation (**SQL Doc**), and exploring object dependencies and visualizing complex databases (**SQL Dependency Tracker**). Because both **SQL Source Control** and **ReadyRoll** support the connected development model, both SSMS and VS can be used for database development.	
Version control	**SQL Source Control** plugs into SSMS and links your databases to any version control system (VCS). The current state of each object in the database is versioned. The VCS contains a folder with the current `CREATE` scripts for SQL Server objects, such as tables and stored procedures.	Working in Visual Studio, database projects are versioned as part of your overall solution. **ReadyRoll** plugs into VS and generates numerically ordered `ALTER` scripts that sit inside the versioned project and gradually take schemas from one version to the next.
Deployment	Users deploy straight from **SQL Source Control**, using **SQL Compare** to compare the source and target databases, and to generate the scripts to synchronize the two states.	Projects are configured in Visual Studio to output a deployment script during build, which can be deployed from Visual Studio or SSMS.
	SQL Multi Script provides a one-to-many deployment capability.	
Automated deployment	**DLM Automation** automates the build of databases via any Continuous Integration (CI) server. It produces a NuGet package to be deployed via automated deployment tools such as Release Management for Visual Studio, Visual Studio Team Services, Octopus Deploy, and Bamboo.	The build of **ReadyRoll** projects can be automated via PowerShell, or any CI server or deployment tool. Native VSTS and Octopus Deploy extensions have been developed to make it even more straightforward.
	Use the **SQL Comparison SDK** to build comparison and synchronization functionality into your own applications.	It is possible to integrate **ReadyRoll** into your own application installers.
Monitoring	**SQL Monitor** provides a web-based overview of SQL Server performance, and alerts users to issues before anyone notices that server performance isn't what it should be. **DLM Dashboard** tracks schema changes across environments and alerts teams to databases that have drifted from their expected state, significantly reducing the risk of deployment errors.	

redgate