The business benefits of database source control

Improving productivity, change management, scalability, and code quality with SQL Source Control
Introduction

For application developers today, it is unthinkable to work without source control. The benefits it brings to software development are so well and so long understood that even lone hobbyist developers will tend to employ a source control system. Database source control was for some time seen as unfeasible, and it is not yet common practice everywhere.

Now that database source control is not merely possible but easy, SQL Developers, Business Intelligence developers, and DBAs are beginning to understand its benefits too. But developers know how source control helps them work better, faster, and more collaboratively. How does this translate into business benefits?

What is source control?

Source control is a system – typically on a centralized server – for tracking and retaining an incremental history of changes to a set of files. In application development, these files are the application source code. Storing code (or files such as architecture diagrams or documentation) in source control allows teams to collaborate on projects, knowing who changed what, when, and why.

Source control (sometimes referred to as “version control”) is the core of what many enterprises call Software Configuration Management (SCM) or Application Lifecycle Management (ALM), where it may also include process and workflow control. In each case, the value is in having a single, authoritative, shared repository of project resources and configurations that helps people work together, and enables monitoring and reporting.

The problem for databases is that version control, SCM, and ALM rest on the sharing and monitoring of a set of files. Database development does not typically involve source files, and so database source control has historically been impossible. Now this is no longer the case, data-intensive business functions can feel the benefit of source control.

How source control helps your business

Management visibility

Source control maintains detailed change histories, and can often be associated with issue tracking systems. For example, SQL Source Control lets you associate database tasks with Microsoft’s Team Foundation Server “work items”.

This means you can easily get a detailed view – or just an overview – of what development work is going on, its progress, who’s doing it, and why.

Compliance and auditing

Your databases and data warehouses are where much – if not all – of your critical business information lives. Having a history of how it’s changed, and making it easy for the teams maintaining it to work efficiently is as important as your backup and recovery strategy.

Regulatory compliance has an impact throughout an organization, from finance departments and CIOs to individual developers and database administrators.

For example, compliance auditors will require DBAs to account for all changes to a database, and detail all those with access to it.
The change tracking provided by source control is the first step to getting your database ready for compliance, and an essential step in maintaining a robust audit trail and managing risk.

**Efficiency and scalability**
Every business wants to reduce its costs. Every growing business wants to ensure that costs (and often head count) don’t have to scale linearly with growth. Genuinely scalable growth is about doing more without spending more, which is where efficient tools and processes come in.

Source control removes the friction of team collaboration, enabling developers to work faster. It affords rapid change sharing, and simplifies change management, making it particularly important for geographically distributed teams. Tools like SQL Source Control simplify deployments, making complex processes more automatable and repeatable.

Because the history it provides is incremental, source control lets developers explore different solutions, and roll back safely in the case of errors. It also makes it easier to implement automated testing. This means that problems can be found earlier, and higher quality code eventually shipped and deployed. That’s inherently desirable, but more than that, it can significantly reduce support and maintenance costs.

**Cost of ownership**
Cost of ownership varies between source control systems. Free tools may have less support, and so higher implementation costs, for instance. Whereas enterprise SCM systems can often be purchased bundled with consultancy and training.

What doesn’t vary is the fact that the easier a system is to use, the quicker it is to learn, to implement, and so to return value.

For database source control, SQL Source Control integrates existing source control systems with the standard development environment, and doesn’t require that developers change the way they work. It’s designed to be simple and easy to use. This means it is easy to adopt, with no additional infrastructure requirements. So the cost of ownership stays low.

**Conclusions**
Implementing source control for databases helps developers, DBAs, and Business Intelligence professionals work more efficiently, which ultimately saves money and expands their potential to do high quality work.

Database source control hasn’t been possible for a long time. Because problems with no feasible solution are readily regarded as a sunk cost, the disparity with application development and the inefficiency caused is often overlooked.

Now database source control is simple to implement, it is no longer necessary to accept that compromise. Database development can enjoy the same benefits as application code, and businesses can start saving money.