

Source Control – Going up the maturity curve

Most of us reading this have had some discussions in and around source control, and some of us interact with a version of it on a regular basis. However, sometimes businesses need a helping hand to get them on a more formal source code control path to allow for all its benefits to flow through their ecosystem.

Here's how we are helping one valuable client uplift their disparate code base into a managed source control.

As a general best practice, all source code should reside in some form of managed source control system. It provides numerous benefits such as:

- allowing multiple developers to work on the same code base
- allowing developers to commit and merge code without conflicts
- letting developers edit shared code without unknowingly overwriting each other's work
- protecting your code in a remote location
- allowing you to revert to code before unwanted changes or mistakes are made
- allowing you to support multiple versions of the code (if needed)
- giving you the freedom to experiment as you can always revert to the last stable version
- giving you the visibility into what changed, when and why (via the check-in comments)

There are numerous tools and version control options out in the market, but one of the best known is git.

As part of our client engagement, it became clear that there was a need for the source files and projects for all the SQL Server BI Stack artefacts and Power BI artefacts to be put under a more formal form of source control. This was due to numerous factors, including that the original development team was being reorganised so that there was no longer one SME responsible for an area, the deliverables the development team were responsible for had changed, the inclusion of multiple teams working on the same objects and the general growth in maturity of the environment as a whole.

So, where to begin?

It was decided, given the starting point of having the current source code located on laptops, servers, OneDrive and SharePoint, that the upliftment and subsequent downstream code release process would be done in two phases. The first phase would be the upliftment of the code base into central repositories, while the second would look at the release management process.

The client had Azure DevOps available that allowed for the creation of git-based repositories. There were also a few client resources that had some experience of how to use git repositories with more advanced features such as the integration of code changes into release pipelines, although not specifically in the BI world.

In order to fast track the upliftment of the code base and adoption of the source control process by developers it was decided to leverage the existing capability and expand it to include the BI stack artefacts.

The first step was to set about creating repositories to house the source code.

The driving idea behind the breakdown of repositories was to have them created in terms of 'units of work'. The thinking being that for those reporting structures that were a logical unit there would be only one repository. An example is where reporting was required from one particular source system. The underlying SQL database, SSIS etl packages, SSAS tabular model and the Power BI files relating to this reporting capability would then all reside in their own Visual Studio projects but in one Visual Studio solution, all in one repository.

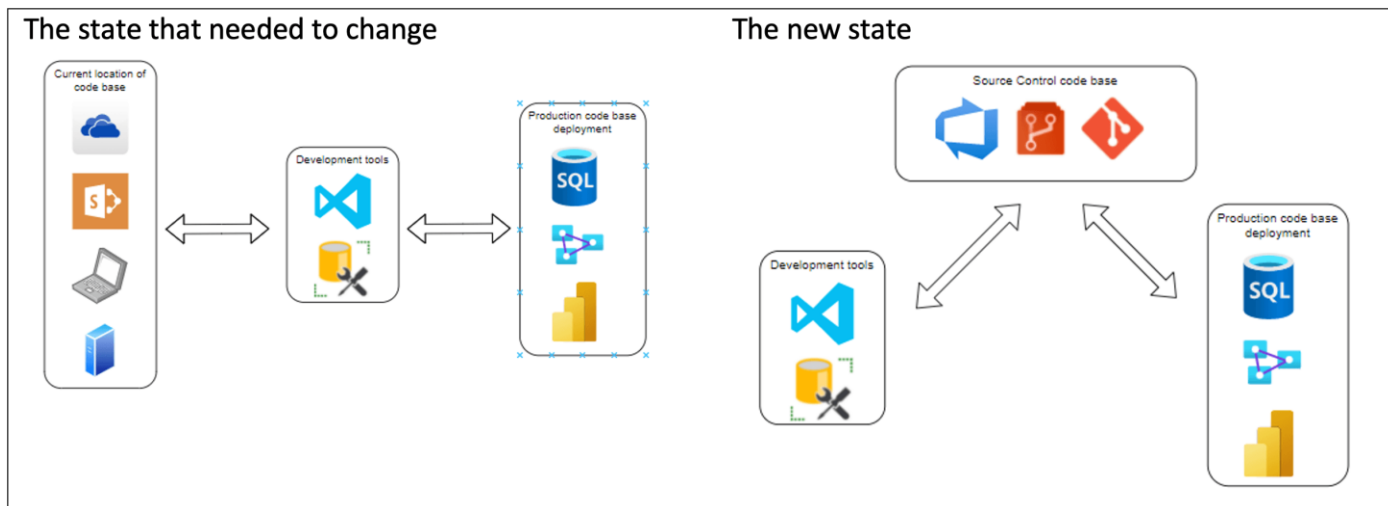
For those artefacts used across a wide range of reporting structures, changes would fit into a narrower field of 'unit of work', and so would warrant its own repository. Here an example would be a SQL database being used as a data warehouse, housed in its own Visual Studio solution.

To facilitate this distinction and to make it easier for developers to find the relevant code base, a naming convention established by the existing capability was expanded on to now include the BI artefacts. So generic, technology agnostic names such as datamodel (SSAS projects), etl (SSIS projects) and database (SQL database projects) were added into the naming convention.

A few training sessions were held for all the developers explaining the new source control mechanism and the new processes in place. This ensured everyone knew what was expected, and to provide the opportunity for questions and answer sessions. Wiki pages and general repository knowledge web links were also made available to ensure that any new projects had a place to go for reference material and guidance to set up the new code base.

In summary, the steps followed for the first part were to:

1. Decide on the focus area
2. Determine a naming convention
3. Create the repositories
4. Train the developers



The second part of code base upliftment deals with the way the code is actually released into production. The current method was for the developers to simply push changes from their various code base locations, after some checks were performed, directly into production.

With the upliftment of the code base into repositories and the desire to move towards a CI/CD deployment process, this process necessarily needs to be updated.

This second phase, currently underway, relies on the use of the new code repositories and involves setting up a code release process that allows for CI/CD changes to be directly deployed to production.

This release process will use:

- an easy to use and understand branching and merging strategy
- repository settings within Azure DevOps, such as branch policy settings, to set up branch level rules
- rules for code review and pull request approvals
- rules for build and integrated testing
- pathways to deal with
 - priority incident management
 - and BAU break/fix work

All the steps ensure the release management process is streamlined.

The ultimate goal is to provide an end-to-end process, that works seamlessly across teams and deployment environments, with sufficient controls in place to help minimise defective code, while allowing for production ready code to be released as soon as it is available.

To see how we can help your business with their source control, please contact a Fujitsu Data & AI specialist now.

Contact

Fujitsu Data & AI
+61 3 9924 3000

© Fujitsu 2022. All rights reserved. Fujitsu and Fujitsu logo are trademarks of Fujitsu Limited registered in many jurisdictions worldwide. Other product, service and company names mentioned herein may be trademarks of Fujitsu or other companies. This document is current as of the initial date of publication and subject to be changed by Fujitsu without notice. This material is provided for information purposes only and Fujitsu assumes no liability related to its use.