



CASE STUDY | MACMILLAN

" It seems like a magic bullet. "

How Redgate's DLM Workshops make Continuous Integration for databases easy to understand – and simple to apply

redgate
ingeniously simple

Enter Macmillan Cancer Support

**WE ARE
MACMILLAN.
CANCER SUPPORT**

Founded in 1911, Macmillan Cancer Support is one of the largest charities in the UK and, in 2014, supported an estimated 5.4 million people affected by cancer through at least one of its services.

Those services were accessed a total of nine million times, and Macmillan health and social care professionals directly supported more people in 2014 than ever, helping around 703,000 people affected by cancer.

As can be imagined, the IT services that support such a big enterprise are complex and plug into every aspect of the charity's work, from fundraising and back office operations through to providing high-quality cancer information online, and hosting an online community where people can chat and support each other.

Enter Continuous Integration



Jane Palmer, the SQL-DBA at Macmillan Cancer Support, wants to introduce Continuous Integration (CI) for the SQL Server databases to make it easier for everyone to work together.

As she says: "In order to work together, you need good tools. You need to be using the same tools, you need the tools to work together without causing conflicts, and, crucially, you all need to be using the tools properly."

"We have some source control in place, but it often feels that developers are stepping on each other's toes, that we're not always testing as robustly as we might, and that our databases are out of step between environments. As DBA, I'm running scripts and copying databases from Live to Test to Dev more than I might otherwise like."

She suspected that CI might be the answer. "It seems like a magic bullet. CI is the practice of merging and testing all developer working copies with a shared mainline several times a day. It means that everything should stay inline, that no one steps on anyone's toes. That all should be shiny and lovely."

Enter Redgate DLM Workshops

Jane Palmer knew what she wanted to do, but wasn't sure where to start. Which is when she turned to Redgate. She had already heard about Redgate's Database Lifecycle Management (DLM) Workshops, which were introduced in 2015. Led by expert trainers from Redgate, or consultants like Northwest Cadence and the DevOpsGuys, the workshops cover the three areas that make DLM possible: source control, continuous integration, and automated database deployment.

Jane chose to attend the CI workshop in London and spent a day in the company of Redgate experts and fellow DBAs and developers, exploring the four stages of CI:

1. Linking the source control repository to a build system
2. Triggering database builds automatically when changes are made
3. Using tools like Redgate's DLM tools to test the database changes
4. Understanding package-based deployments of databases

"By the end of the day," Jane Palmer says, "I could see how CI was supposed to work in the world of databases. From a standing start, we looked at how to link SQL Source Control to a version control system like TFS. It works with pretty much any system, and DLM Dashboard (Redgate's super, free product which I had a lot of fun testing last year) helps keep track of database changes. For me, the most important part was how to automate the process."

She took the knowledge back to work with her and immediately saw the difference the workshop made. "I now have the confidence to apply source control properly, deploy to the Live database when I want to – and roll the deployment back if I need to. I'm spreading the knowledge through the team and we'll be introducing true CI in the months ahead."



For further information about the invaluable work **Macmillan Cancer Support** carries out, please visit them online.

For further information about Redgate's DLM Workshops, please **visit the Redgate website**.

And if you'd like to read more about Jane Palmer's adventures as a DBA, you can **read her blog online**.

