

Put databases under source control

Alessandro Alpi



Sponsor & Media Partners



November 9th, 2013



#257 | ITALY 2013 - VERONA



#sqlsat257
#sqlsatverona

Organizers



November 9th, 2013



#257 | ITALY 2013 - VERONA



#sqlsat257
#sqlsatverona

About me

- MVP
- MCTS
- MCITP
- MCP
- MCT

- Ita blog: <http://blogs.dotnethell.it/suxstellino>
- Eng blog: <http://suxstellino.wordpress.com>
- Twitter: @suxstellino
- Linkedin: it.linkedin.com/in/suxstellino/

Agenda

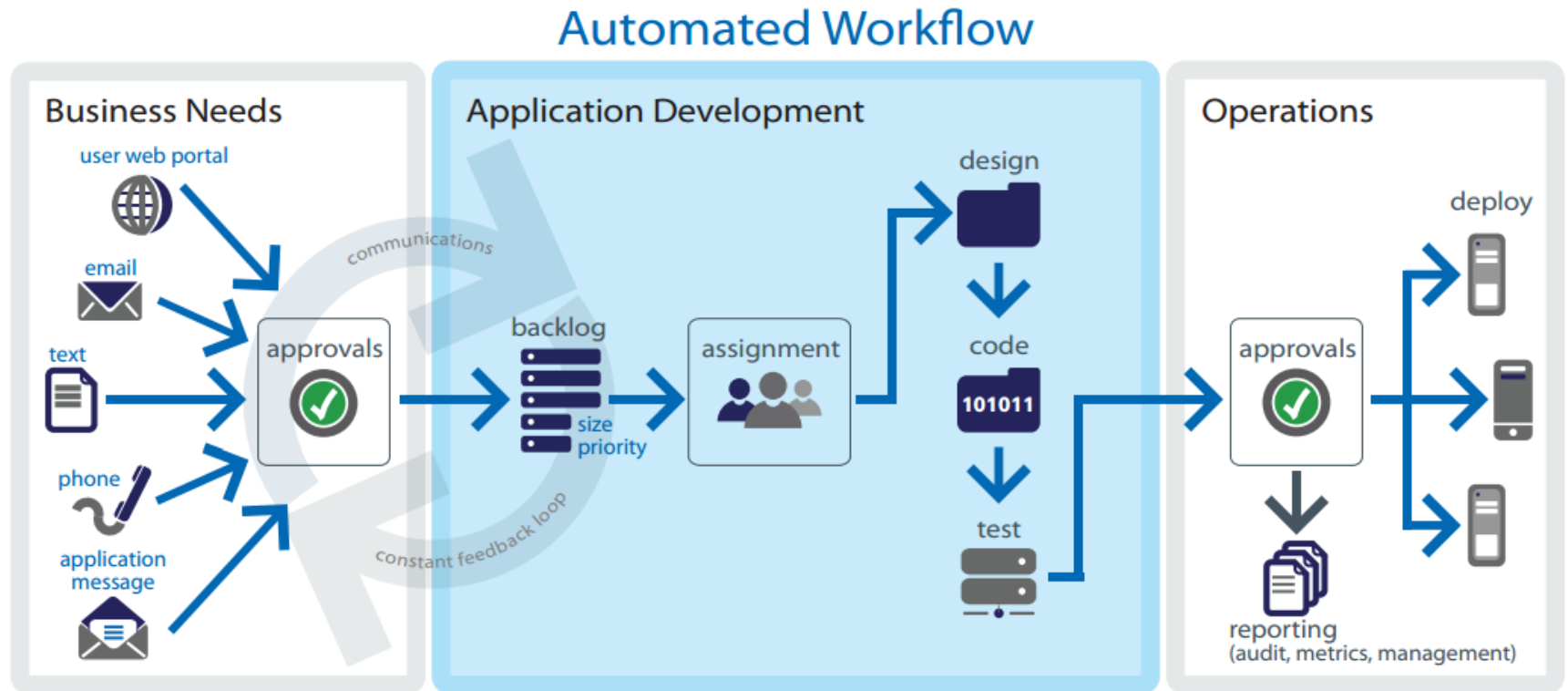
- Concetti di ALM
- Source control manager
- Database vs Codice
- Tools per database
- Operazioni sui progetti
- Tools per gestire le operazioni
- Conclusioni
- Q&A

Che cosa si intende con ALM?

Application Lifecycle Management (ALM) rappresenta l'unione di attività di gestione di business con attività di ingegneria del software, resa possibile dall'utilizzo di strumenti che facilitano la gestione delle fasi di: analisi dei requisiti, progetto architetturale, sviluppo, testing, gestione delle release, del change e del deployment.

(fonte Wikipedia)

Che cosa si intende con ALM?



Perchè ALM?

- Rottura delle barriere tra i team (integrazione)
- Rilascio di software di qualità
- Rilascio di software in tempi brevi
- Soddisfazione del cliente
- Migliore organizzazione del lavoro
- Monitorizzazione e tracciabilità delle attività
- Migliore gestione del codice (più «pulito»)

ALM e database

- La parte di database ha analisi e sviluppo
- I database devono poter essere distribuiti
- I database devono essere sincronizzati nell'ambiente di sviluppo
- Il database avrà «cambiamenti» da associare ad «attività»
- Il database dovrebbe essere testato
- Di certo è una cosa di cui fare deploy 😊

Soluzioni e tool – development

- Management Studio non basta
- Visual Studio e Data Tools
- Integrazioni di tool di terze parti con SSMS (esempio: Red-Gate SQL Source Control)
- Software di terze parti non integrati con IDE proprietari
- Li vedremo più avanti..

Source Control Manager

- Gestore delle versioni, dei cambiamenti del nostro codice (e non solo)
- Entità condivisa nelle fasi di sviluppo, deploy e team management
- Dotato di interfaccia (anche grafica)

SCM – Perché li utilizziamo

- Versioni del nostro codice
- Salvataggio sicuro dei nostri file
- Distribuzione delle linee di sviluppo al team
- Creazione di un punto centrale per i deploy
- Automatizzare processi di build e di test
- I bisogni di ogni team..

SCM – Parlando di database

- Il DB può essere un file «nell'applicazione»
- Il DB «sta su server»
- Il DB persiste dati utente
- Il DB non è tutto e solo codice
- Tuttavia i cambiamenti al DB devono riflettersi a tutto il team

Il source control potrebbe sembrare «scomodo»

Ma senza un SCM

- Come potremmo **semplicemente** gestire le fix?
- Come prevenire regressioni?
- Come avere **velocemente** più ambienti di sviluppo?
- Come creare **semplicemente** una nuova linea dev?
- Come utilizzare versioni differenti dello stesso DB?
- Come potremmo rendere il DB sincronizzato agli ultimi cambiamenti dell'applicazione?

DB vs. codice – così diversi?

- Il database è comunque codice (programmabilità, ddl, grant, ecc.)
- Le tabelle di «dominio» sono come tanti enum (dati statici)
- Anche il DB dovrebbe essere trasformato in più line di sviluppo durante le operazioni di branch del codice

DB vs. codice – così diversi?

- I puntamenti ai linked server sono configurazioni (come l'app.config)
- Le server login sono configurazioni di ambiente
- Il database **persiste i dati**. Non è un problema **da source control**

Perché mettere il DB sotto SCM

- Versioni dei nostri oggetti (DDL) e della programmabilità su database
- Le label comprendono il database, in modo da poter tornare ad una situazione precedente
- Team sincronizzati sulla get di una versione (tipicamente l'ultima)
- Per fare versioning anche dei dati statici

E ancora..

- Continuous Integration (con test)
- Branch (più linee di sviluppo e più contesti)
- Ambienti isolati per team dislocati
- Atomicità tra applicazione e DB
- Salvataggio della documentazione del database

SCM – Eccone alcuni

- TFS (on-premises e «service»)
- Git
- Mercurial
- Subversion
- CVS
- Perforce
- ...

Tool per la gestione SCM per DB

- Visual Studio
 - SQL Server Data Tools
- Red-Gate Source Control
 - SQL Test (for CI)
- ApexSQL Versions
- ...

Operazioni possibili con SCM

- Alcune delle operazioni sono:
 - Get
 - Commit/Checkin
 - Undo
 - Save (working folder)
 - Delete (working folder)
 - Edit (working folder)

DEMO

Connessione al TFSservice tramite VS Team explorer e Red-Gate Source Control

Visual Studio + Data Tools

- Connected database development

CONNECTED DEVELOPMENT



Visual Studio + Data Tools

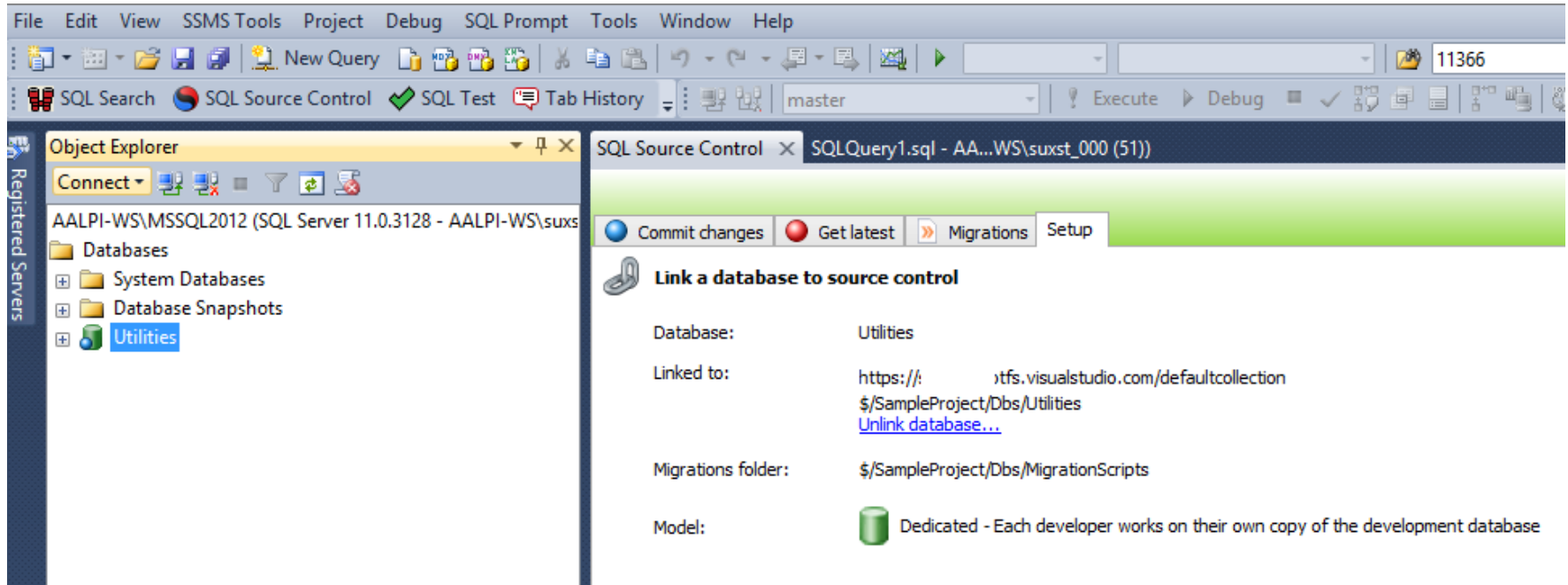
- Project based development

PROJECT BASED DEVELOPMENT

 1 st class T-SQL Support	 Isolated	 Integrated	 Schema Insight
Go To Definition	Source Code Based	Source Code Control	Point-in-time Snapshots
Find All References	F5 Debugging & Testing with LocalDB	MSBuild	Visualize Schema Differences
Refactoring		Headless Command Line Tools	T-SQL Static Code Analysis

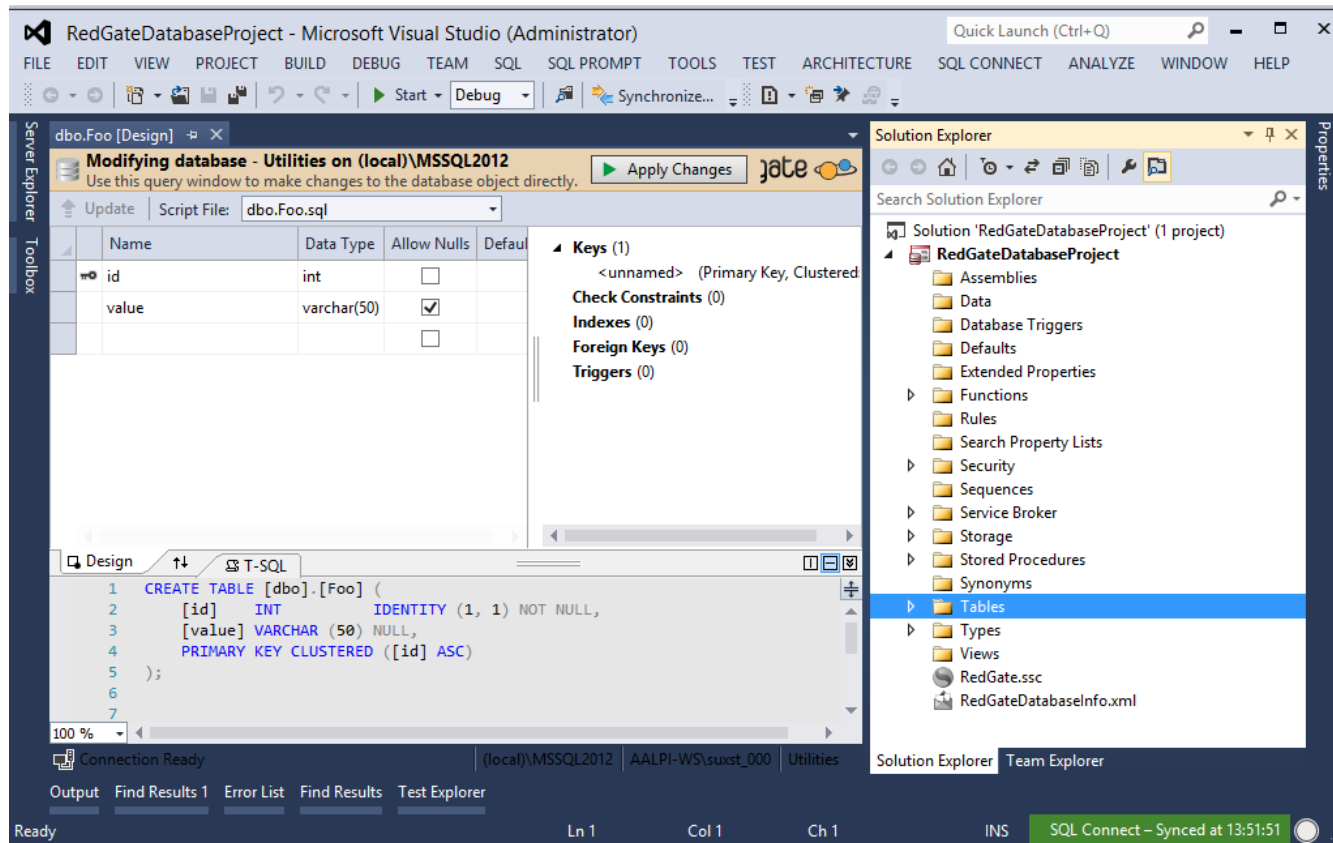
Red-Gate SQL Source Control

- Integrazione con SQL Server Management Studio



Red-Gate SQL Source Control

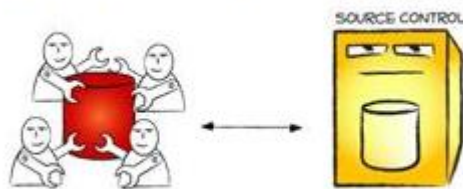
■ Integrazione con Visual Studio (SQLConnect)



Red-Gate SQL Source Control

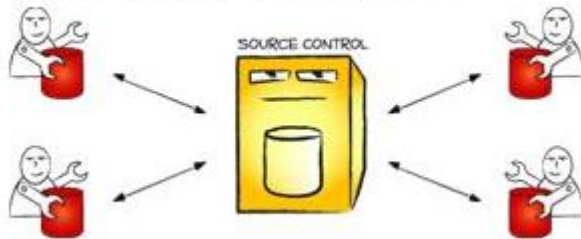
- Modello di sviluppo condiviso

Shared database development



- Modello di sviluppo dedicato (consigliato)

Dedicated database development



Operazioni possibili da SCM

- Con Red-Gate SQL Source Control è inoltre possibile:
 - Connettersi ad una working folder
 - ci si muove con il Team Explorer di Visual Studio
 - Connettersi direttamente al SCM
 - il tutto è gestito direttamente da Management Studio
 - Salvare i dati statici sotto source control
 - Automatizzare il processo di refactor con migrazione

Il Team Explorer

Indipendentemente dal tool che si usa Team Explorer consente:

- Migliore gestione dei changeset
- Migliore associazione dei changeset ai task
- Miglior controllo sulle fasi di commit e di review
- Gestione centralizzata delle policy di checkin
- Single point per la gestione del team project

DEMO

Gestione di un database e di un team project utilizzando TFSservice e i tool per la gestione del SCM

Conclusioni

- Quali tool utilizzare?
 - Ogni tool ha le sue peculiarità
 - Red-Gate SQL Source Control consente di gestire i dati in una maniera molto semplice
 - Visual Studio garantirà la stessa struttura del progetto database
 - Visual Studio è più comodo per gli sviluppatori (o SQL Connect)
- Quali parametri dovremmo considerare?
 - Com'è il nostro team
 - Quali sono i requisiti minimi per la scelta del source control tool
 - Quanto posso spendere?
 - Posso permettermi la curva di apprendimento se cambio IDE?
- In definitiva però, il Source Control dovrei usarlo 😊

Resources

- <http://www.getlatestversion.it/> (ALM community italiana)
- <http://blog.knodev.com/search/label/ALM> (blog di Davide Vernole su ALM)
- <http://www.codewrecks.com/blog/> (blog di Gian Maria Ricci su ALM)
- <http://mattvsts.blogspot.it/> (blog di Matteo Emili su ALM)
- <http://www.codinghorror.com/blog/2006/12/is-your-database-under-version-control.html>
- <http://odetocode.com/blogs/scott/archive/2008/01/30/three-rules-for-database-work.aspx>
- <http://odetocode.com/blogs/scott/archive/2008/01/31/versioning-databases-the-baseline.aspx>
- <http://odetocode.com/blogs/scott/archive/2008/02/02/versioning-databases-change-scripts.aspx>
- <http://odetocode.com/blogs/scott/archive/2008/02/02/versioning-databases-views-stored-procedures-and-the-like.aspx>
- <http://odetocode.com/blogs/scott/archive/2008/02/03/versioning-databases-branching-and-merging.aspx>
- <http://www.red-gate.com/products/sql-development/sql-source-control/>
- <http://vsaralmassessment.codeplex.com>
- http://it.wikipedia.org/wiki/Application_lifecycle_management

Q&A

- Domande?

November 9th, 2013





#sqlsat257
#sqlsatverona

THANKS!

November 9th, 2013



#257 | ITALY 2013 - VERONA



#sqlsat257
#sqlsatverona