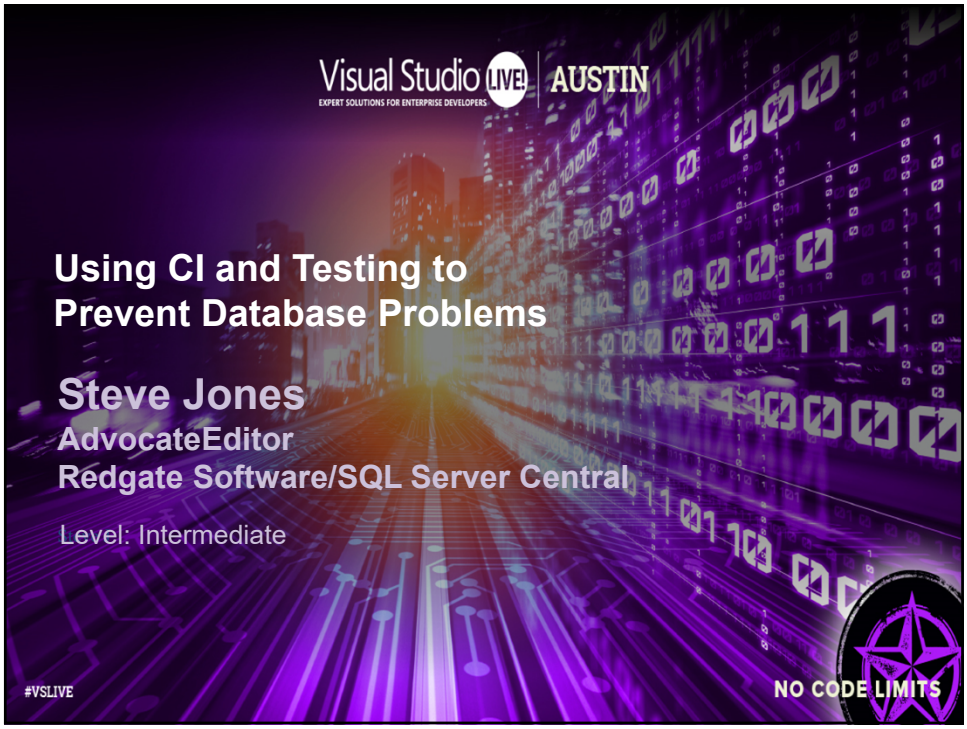


Goals

- Understand how the database development integrates with CI
- Limitations of database work that must be accounted for in a CI process
- Tips that help you begin to integrate CI into your database development

Do feel free to ask questions!

Visual Studio LIVE!

Steve Jones

Advocate, Redgate Software
Editor, SQLServerCentral
he/him

30 years SQL Server data experience

DBA, developer, manager, writer, speaker in a variety of companies and industries

Founder, SQL Server Central

Currently the editor in chief, with the goal of helping you learn to be a better data professional every day

14-year Microsoft Data Platform MVP

I have been honored to be recognized by Microsoft for the as a Data Platform MVP working with SQL Server

/in/way0utwest @way0utwest sjones@sqlservercentral.com www.voiceofthedba.com

Visual Studio Live!

What is continuous integration?

"Continuous Integration is a *practice* designed to ensure that your software is always working, and that you get comprehensive feedback in a few minutes as to whether any given change to your system has broken it."

Jez Humble, ThoughtWorks, author of
"Continuous Delivery"

Visual Studio Live!

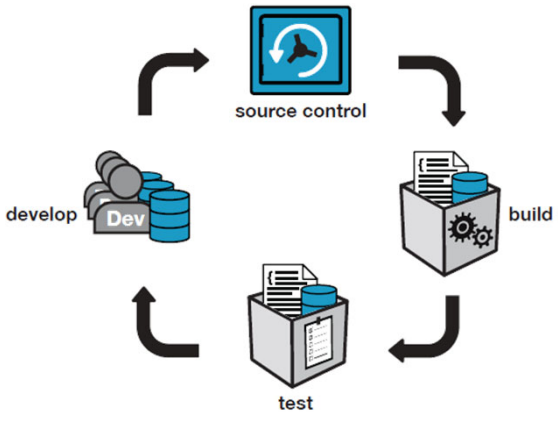
What is database continuous integration?

"Database Continuous Integration is a *practice* designed to ensure that your database ~~software~~ is always working, and that you get comprehensive feedback in a few minutes as to whether any given change to your system has broken it."

Jez Humble, ThoughtWorks, author of "Continuous Delivery"

Visual Studio Live!

Database Continuous Integration



Visual Studio Live!

What is *build*?

- For application code = compile
- For database code = database creation script
  - But only for a new installation!
  - Upgrade scripts required for ***existing*** installations
  - Need to preserve the state of the data

The Database Build

- New Database
  - Ensure the codebase is valid
  - Ensure CI runs fast
  - Remove the challenge of versioning a specific schema
- The DB Build Process
  - Get the code from a VCS
  - Order the code according to the rules of SQL
  - Execute code on the database platform
  - Run tests

Tools we need

- VCS
- Process for capturing database code
- CI Server
- Tooling for database build
- (possibly) test data
- Database Testing Framework
- Process for producing an artifact

Tools we need

- Testing framework
  - We need a way to unit test our code easily.
  - A framework handles known starting state and transaction support
- Choices
  - tSQLt
  - TSQLUnit
  - others

Continuous Integration Setup

Visual Studio Live!

An Integration Database

Contains

- Schema
- Static data

Why?

- A database for the application
- Contains all the latest changes from developers
- First place to test all the team's code

Visual Studio Live!

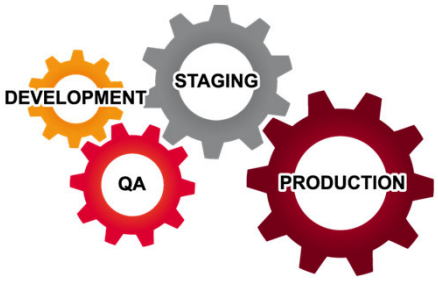
INTEGRATE TESTING IN CI

Why it's important to test database code

Visual Studio Live!

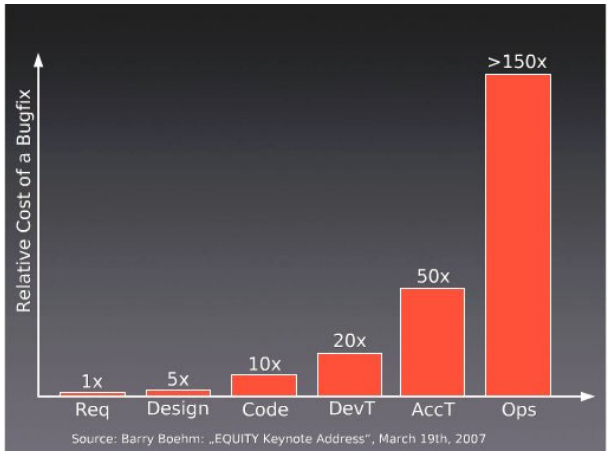
Where does testing happen?

- Testing isn't just done in QA
- Be aware of the cost of fixing a bug



Visual Studio Live!

Cost of Bugs



Visual Studio LIVE!

Does the cost of bugs rise?

If I can fix a bug in my code before it gets to QA, QA has never seen the bug. There's no bug in the bugtracking database, there's no need to review the bug at a weekly cross-functional bug triage meeting, and there's no need to write specific regression tests that specifically make sure that bug is fixed. There's also no need to perform those specific regression tests on every build that follows to make sure it's still fixed. There's no need to hold a meeting to justify the cost of fixing the bug versus the cost of simply leaving it in and documenting its presence and its workaround. Just there, I've saved a ton of time.

The costs explode even higher once the software is in the field. Once it's in the field, it hits support every time the bug is reported in the field (multiple times, usually, as of course level 1 support is usually going to blow off the report, tell them to reboot, or whatever the "filter out bogus complaints" method of the week is). Finally it might bubble up through support but only after it gets seen multiple times, costing us money every time. Then it gets argued about by who-knows-who (more time/money) until finally someone tells development it's a bug, and then we have to hold meetings and decide whether it's important enough to fix immediately, whether it should go in a service pack or just the next version, etc. We have to write up a technical bulletin and distribute it, that bulletin has to be reviewed by documentation, product management, QA, and who-knows-who else. Then QA has to specifically add test cases to make sure the fix is there in future versions, etc.

Visual Studio LIVE!

Does the cost of bugs rise?

If I can fix a bug in my code before it gets to QA, QA has never seen the bug. There's no bug in the bugtracking database, there's no need to review the bug at a weekly cross-functional bug triage meeting, and there's no need to write specific regression tests that specifically make sure that bug is fixed. There's also no need to perform those specific regression tests on every build that follows to make sure it's still fixed. There's no need to hold a meeting to justify the cost of fixing the bug versus the cost of simply leaving it in and documenting its presence and its workaround. Just there, I've saved a ton of time.

The costs explode even higher once the software is in the field. Once it's in the field, it hits support every time the bug is reported in the field (multiple times, usually, as of course level 1 support is usually going to blow off the report, tell them to reboot, or whatever the "filter out bogus complaints" method of the week is). Finally it might bubble up through support but only after it gets seen multiple times, costing us money every time. Then it gets argued about by who-knows-who (more time/money) until finally someone tells development it's a bug, and then we have to hold meetings and decide whether it's important enough to fix immediately, whether it should go in a service pack or just the next version, etc. We have to write up a technical bulletin and distribute it, that bulletin has to be reviewed by documentation, product management, QA, and who-knows-who else. Then QA has to specifically add test cases to make sure the fix is there in future versions, etc.

Visual Studio Live!

Database Code is Long Lived

- Database code is a foundation for applications
- Once code is deployed, it can be hard to change
- It shouldn't be, but often it is
- If we make mistakes here, we live with them

Visual Studio Live!

Mistakes We Make

- Database standards (naming, style, structure)
- Logic errors
- Security errors

What is *test*?

- For .NET code, Nunit
  - Runs on a developer's machine and build server
- What about the database?
  - tSQLt is an open source framework for testing SQL Server databases
  - tSQLt.org
  - Support via GoogleGroups

Testing the Build

Visual Studio Live!

Adding to the test suite

- We can't test everything
- We shouldn't test everything
- Let's add tests as we find issues

Visual Studio Live!

Adding more tests

Visual Studio Live!

Testing Limitations

- Performance is hard to test
  - Hardware
  - Load (other CI or dev process)
  - Data size
- Dependencies outside the database
  - Other databases can be on the build server
  - Must be updated by the development process
  - Linked Servers, ETL, config, etc.

Visual Studio Live!

Testing Limitations

- Some T-SQL Structures
  - Insert .. Exec
  - Synonyms
  - Some transactions (tsqlt starts its own transaction
  - Misc limitations with objects (GitHub Issues)
- Smoke Testing
  - NOT SUITABLE FOR PRODUCTION
  - “fake” functions will cause issues

Visual Studio Live!

The End

Thank you

Fill out Evaluations

Any Questions?

www.voiceofthedba.com
sjones@sqlservercentral.com
@way0utwest
/in/way0utwest

Visual Studio Live!

References

- http://assets.red-gate.com/products/sql-development/assets/continuous-integration-using-red-gate-tools.pdf
- http://www.jetbrains.com/teamcity/
- http://developers.slashdot.org/story/03/10/21/0141215/software-defects---do-late-bugs-really-cost-more
- http://tech.lds.org/index.php?option=com_content&view=article&id=238:the-cost-of-bugs&catid=1:miscellanous
- http://www.manageware.co.il/solution/portfolio/auto-deploy/