

Sergey Olontsev

SQL Server MVP, MCM.

sergey@olontsev.biz

[@SergeyOlontsev](https://twitter.com/SergeyOlontsev)

<http://olontsev.ru/>

<http://SergeyOlontsev.com/>



DevOps for Databases

With examples in MS SQL Server.

Sergey Olontsev

Software Expert,
Kaspersky Lab

SQL Server 12+ years.
ETL, HighLoad and BigData, performance
tuning.

sergey@olontsev.biz
[@SergeyOlontsev](https://t.me/SergeyOlontsev)
<http://olontsev.ru/>
<http://SergeyOlontsev.com/>

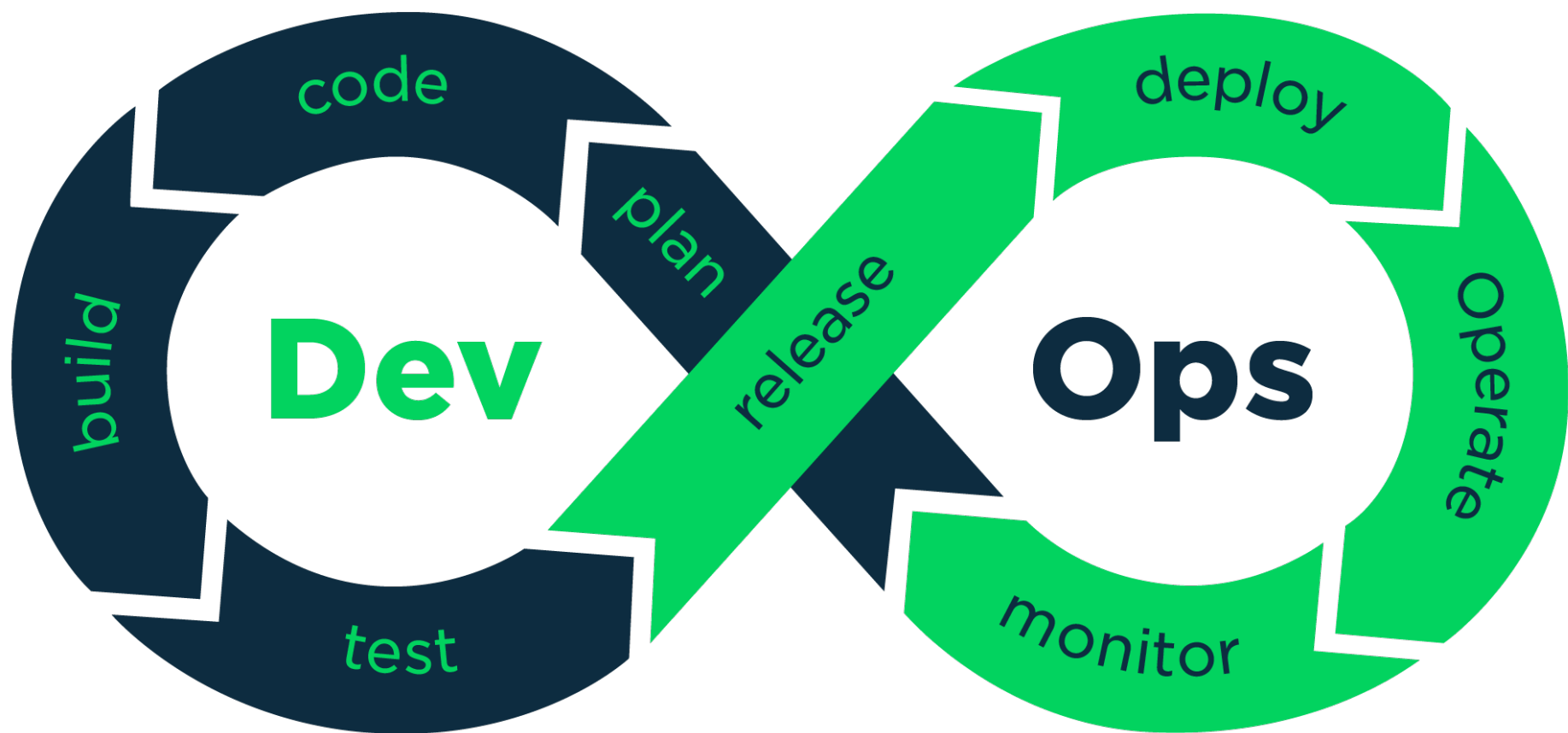


Microsoft
CERTIFIED
Solutions Master
Data Platform

Microsoft
CERTIFIED
Master
SQL Server 2008



DevOps



CI & CD

Continuous Integration (CI) is a development practice that requires developers to integrate code into a shared repository several times a day. Each check-in is then verified by an automated build, allowing teams to detect problems early.

By integrating regularly, you can detect errors quickly, and locate them more easily.

Continuous Deployment (CD) is closely related to Continuous Integration and refers to the release into production of software that passes the automated tests.

Database vs. Application



Database Structure & Code

```
create table [dbo].[Orders] (  
    [OrderId]      int not null,  
    [CustomerId]   int not null  
);  
go  
  
create procedure [dbo].[GetOrders]  
as  
begin  
    set nocount on;  
  
    select  
        [OrderId],  
        [CustomerId]  
    from [dbo].[Orders];  
end;  
go
```

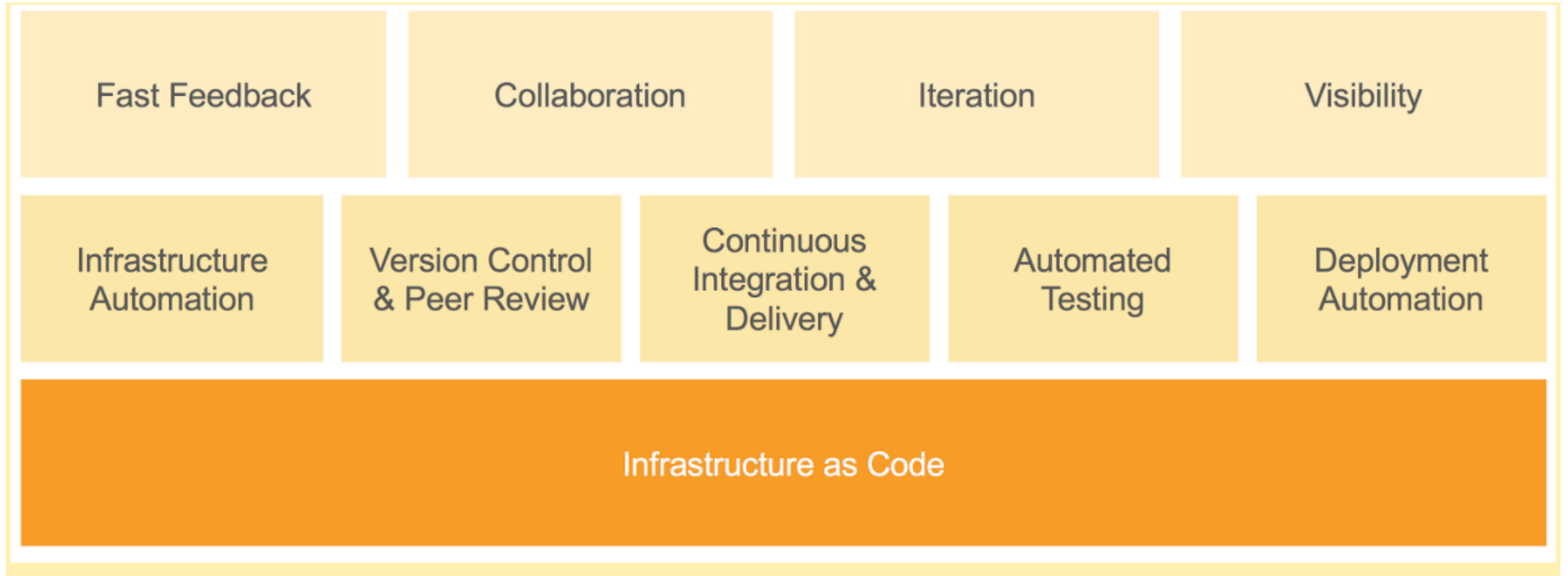
DBA



Database Configuration & Other Stuff

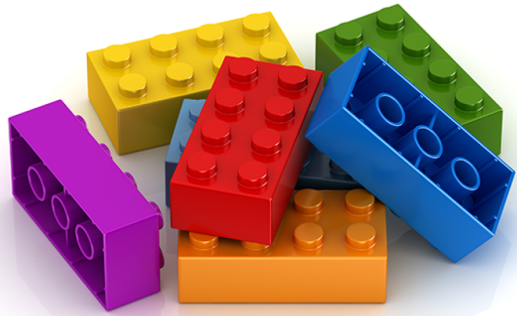
- Security permissions
- Server configuration
- Server level objects (linked servers, jobs, custom errors, etc.)
- Database Indexes & Maintenance Plans
- Backups

Infrastructure as Code



Database Testing

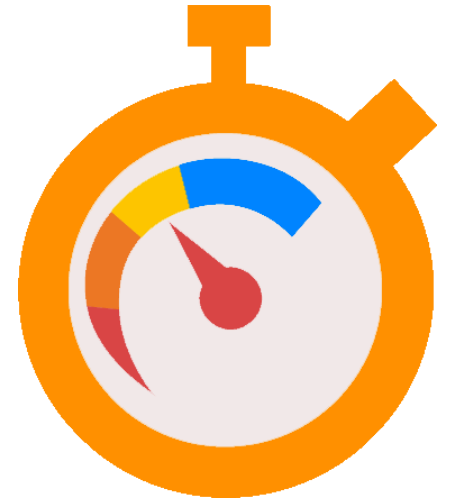
Unit



Integration



Performance



What is TDD?

Test-first development 😊

Write enough code to pass tests

Unfortunately, not so widely used by database developers.

Some Concepts

All tests should be automated and easily re-runnable by any developer in your team.

Probably “one-click solution” or even fully automated one.

Either integrated in your IDE (Visual Studio) or are created using third party framework

It could be even stored procedures or powershell scripts written by you.

Benefits

- Easy and quick code checking for correct behavior in specified use cases.
- Simple ensuring that nothing is broken when adding new feature or changing logic.
- Helps you easily change code written by other team members.

Why still not so used?

- Lack of knowledge.
- No time or desire to learn a new tool.
- It takes a lot more time to write tests and prepare the infrastructure.

Performance Considerations

- Your tests are just code that runs inside your database.
- Try not to clear and insert test data before each test.
- Choose tools that allow you to reuse tests code (for example, executing a series of stored procedures one by one).

Unit Testing



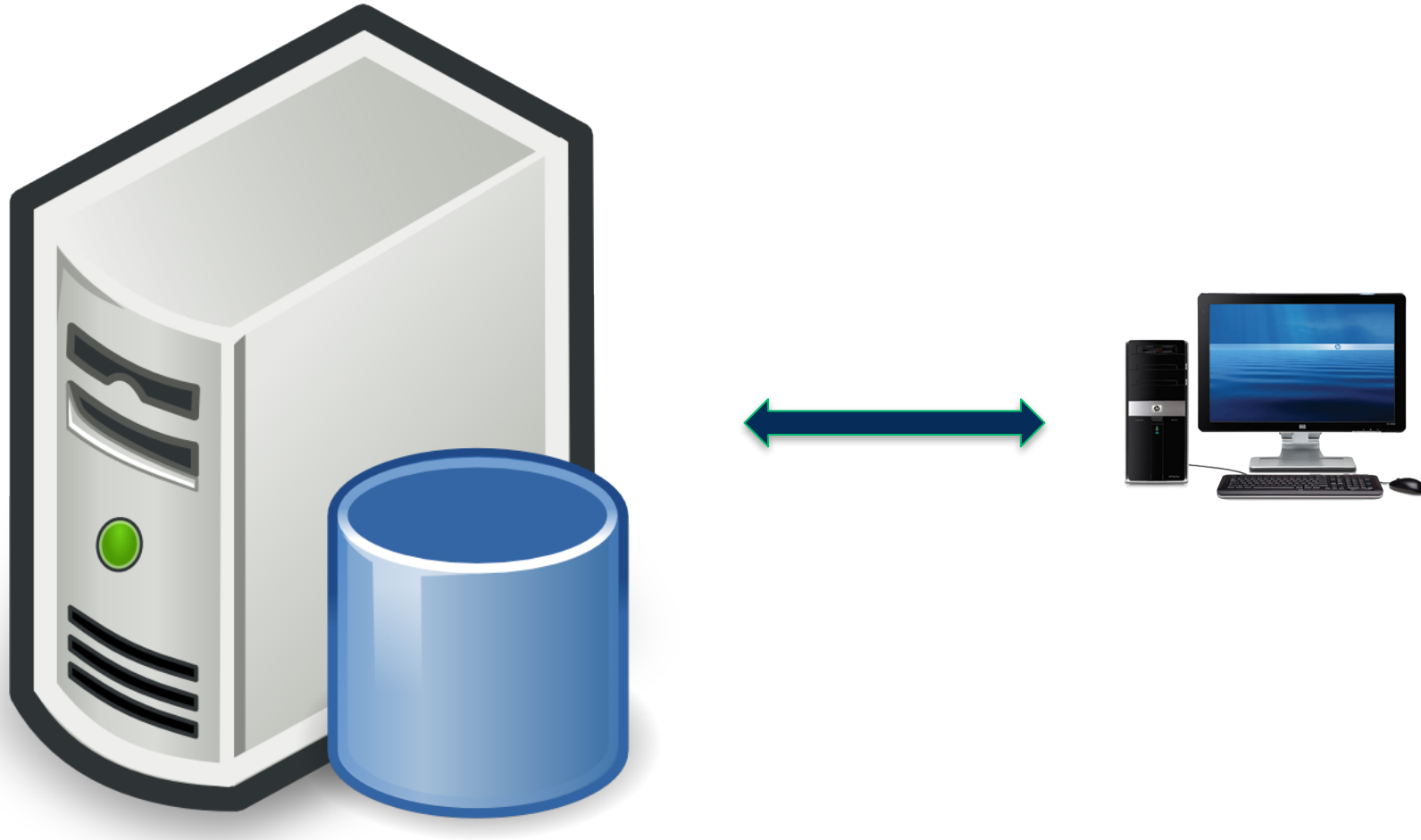
Not a Perfect Approach

- You cannot imagine all the scenarios that could happen in the real world. But you'll definitely reduce the amount of bugs.
- Create integration testing solutions to check how your system works as a whole.
- Load testing will help you to check the performance and behavior of critical queries.

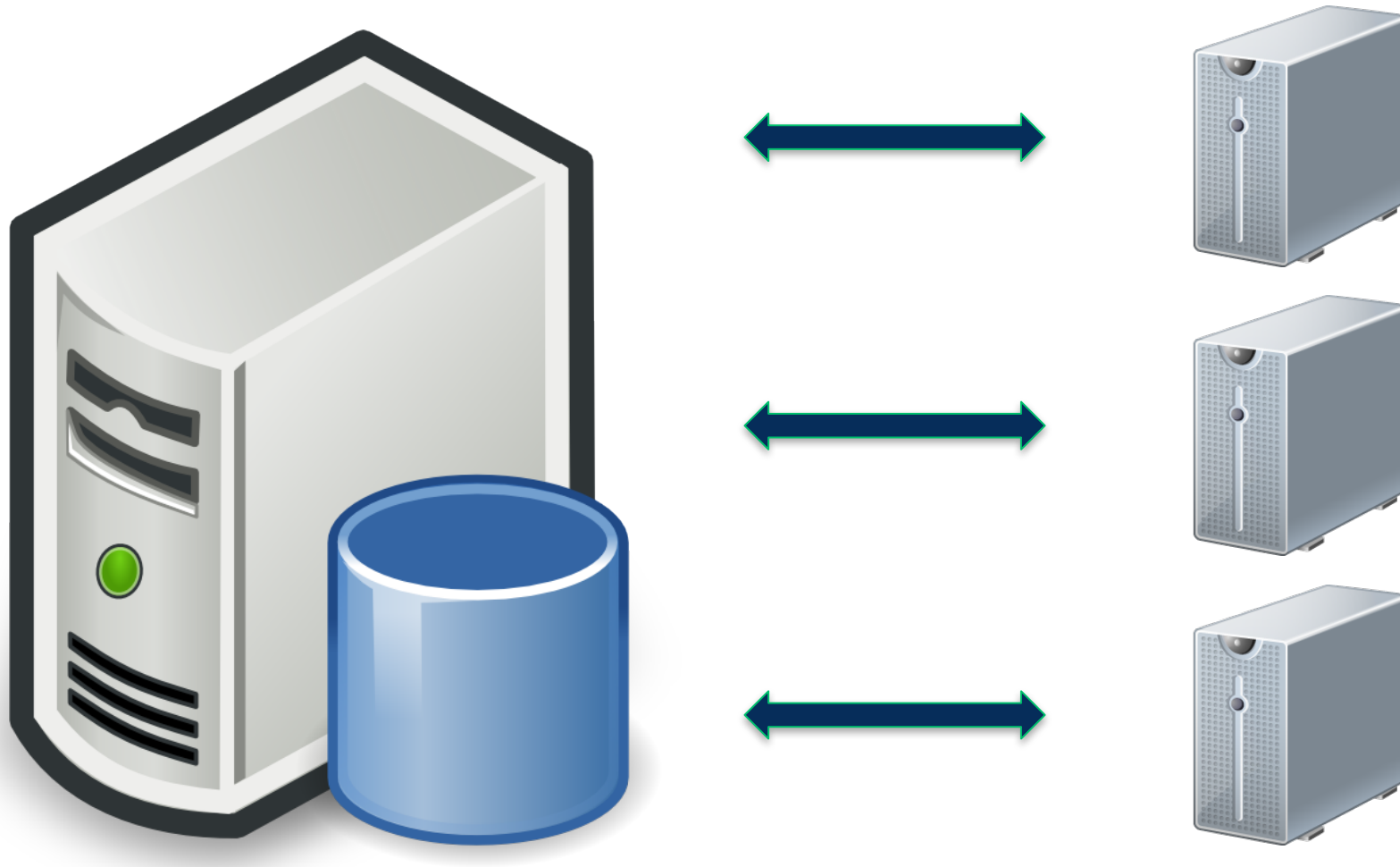
Continuous Integration

- Each commit triggers an automatically deployment and full test circle.
- Don't commit small changes to the common repository.
- Run simple tests on your local machine first.
- Available in TFS, Team City, etc.

Performance Testing – The Wrong Way



Performance Testing – The Right Way





SSDT Demo

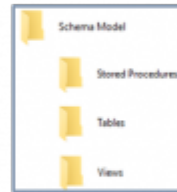
New Version

V1

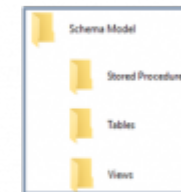


V2

```
CREATE TABLE Orders (  
  OrderId int,  
  CustomerID int, )
```



```
CREATE PROC GetOrders AS  
  SELECT OrderId, CustomerID  
  FROM Orders
```



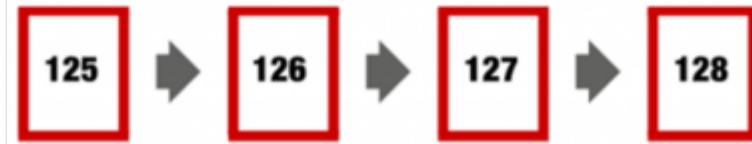
```
CREATE TABLE Orders (  
  OrderId int,  
  CustomerID int,  
  LastEdited datetime2 )
```

```
CREATE PROC GetOrders AS  
  SELECT OrderId, CustomerID,  
  LastEdited  
  FROM Orders
```

Generating Deployment Scripts

V1  **V2**

**ALTER TABLE Orders ADD
COLUMN LastEdited datetime2**

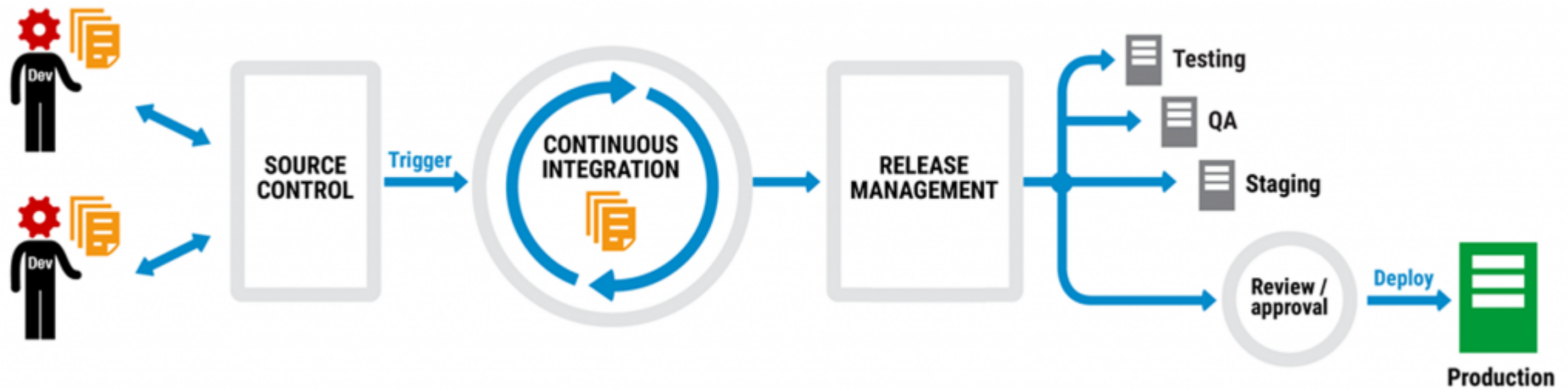


**ALTER PROC GetOrders AS
SELECT CustomerID, LastEdited
FROM Orders**

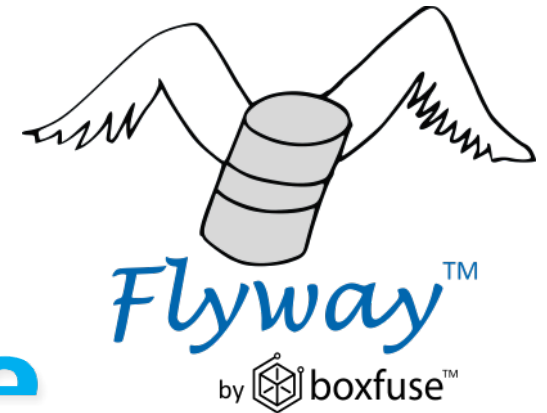
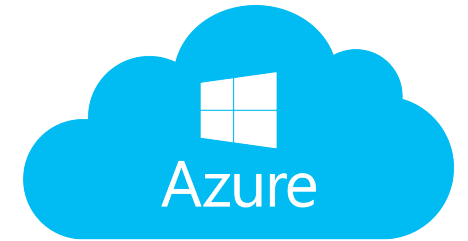
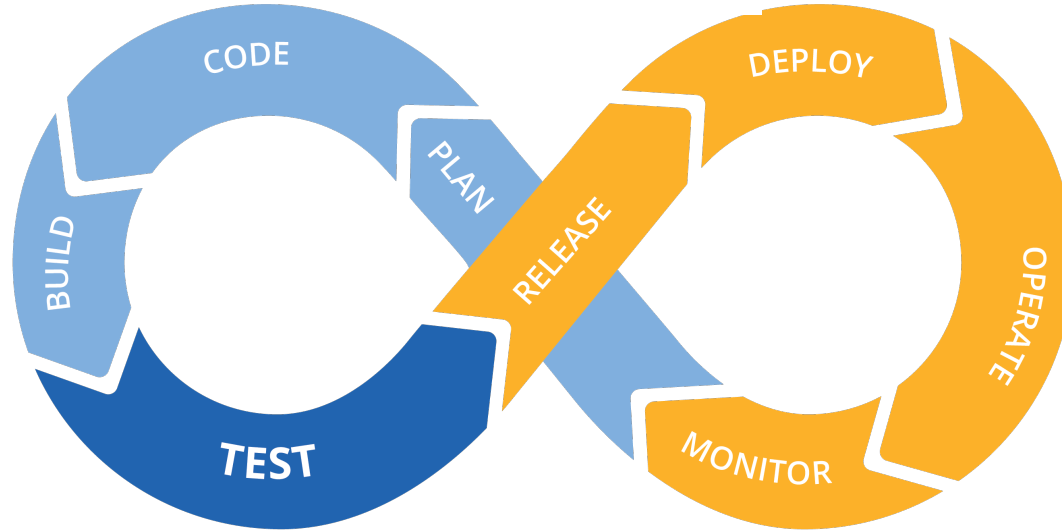
DevCon 2013



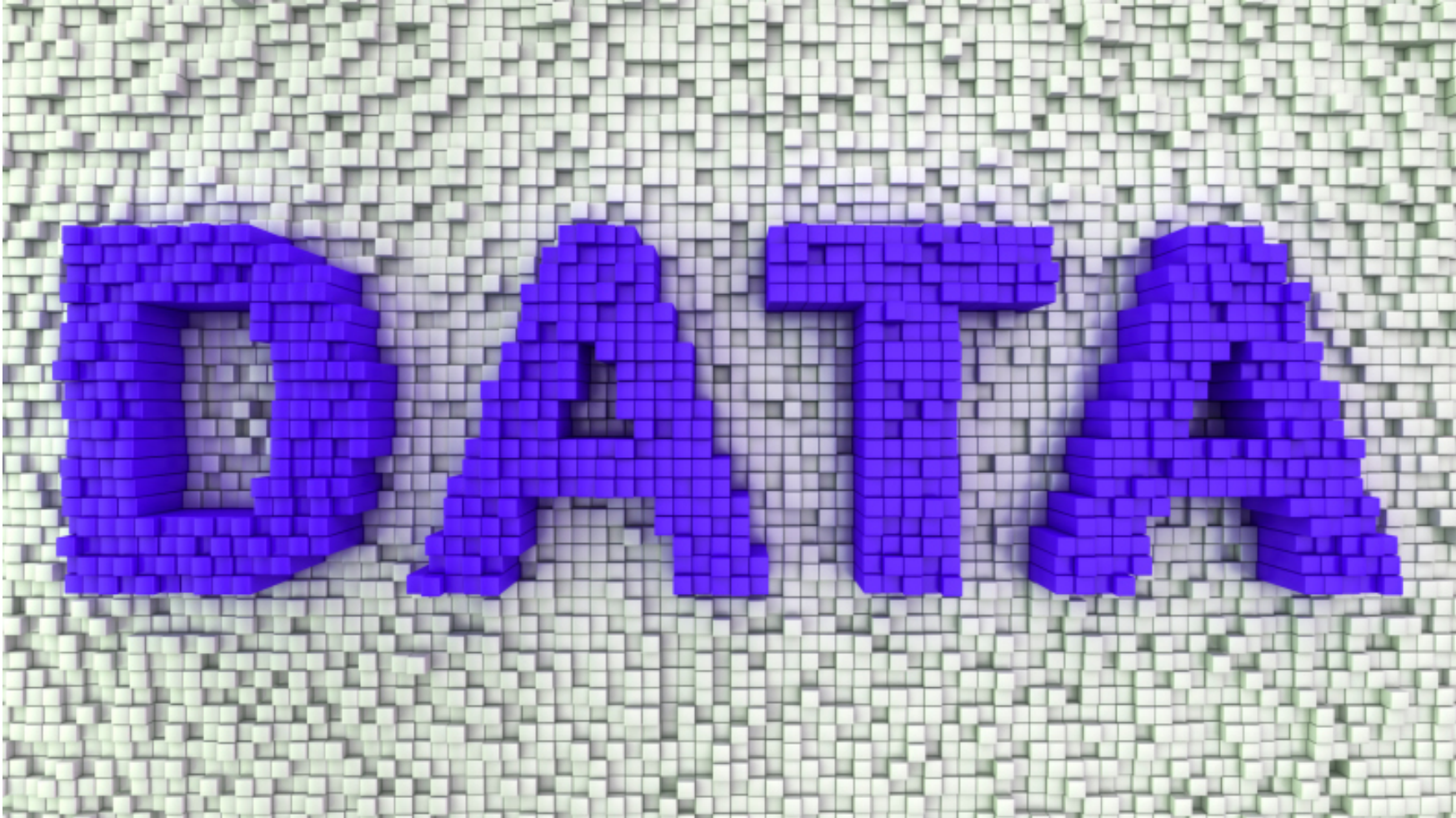
Deployment Pipeline



Tools



One Key Difference



Questions?

