Database DevOps

Strategies to Address DevOps’ “Last Mile”
Scott Ambler

• Helps enterprise-class organizations around the world to improve their processes and organization structures
• Thought leader of:
  – Agile Modeling (AM) method
  – Agile Data (AD) method
• Co-creator of the Disciplined Agile (DA) framework
• Author of 20+ books
• scott [at] scottambler.com
• Advisory Board, ScaleFree
• Advisor, SEMAT
We’re going to cover a lot of ground
Agenda

• Disciplined DevOps
• Organizational Challenges
• Strategies for Database DevOps
• Parting Thoughts
Disciplined DevOps
A Good Start....
A Better View: Agile Delivery + Operations

- Multi-modal approach to software development
- Operations activities are also streamlined and “leaned out”
- Some teams will adopt a “you build it, you run it” philosophy, but a common operational infrastructure still required
The BizDevOps Vision
The DevSecOps Vision
The Database DevOps Vision
Explicit Release Management and Supports
Our View: Disciplined DevOps
Exercise: Database DevOps in Your Organization

- Within your team, discuss the following questions:
  - What aspects/challenges of your current situation would you like to move away from?
  - What would you like to move towards?

- Capture one item per sticky note
  - Up to 3 “FROMs”
  - Up to 3 “TOs”
Cultural Impedance Mismatch

- Developers
  - Have embraced evolutionary, and now agile and lean, strategies for over two decades
  - Are generally weak on data skills
- Data professionals
  - Just now starting to embrace agile strategies for DW/BI
  - Are generally weak on development skills

Source: Agiledata.org/essays/culturalImpedanceMismatch.html
Big Requirements Up Front (BRUF) Has Failed
Feature Usage Within Deployed Applications

- Detailed requirements specs increase project risk
- Traditional “change mgmt” approaches are little more than “change prevention” strategies
- Stakeholders want software which fulfills their needs, not software that fulfills a spec
- The real goal is to understand and then implement requirements, not document them

Source: Chaos Report v3, Standish Group

© Scott Ambler + Associates
Current State: Database Testing

• 88% of organizations considered data to be a corporate asset

• Of those that do, only:
  – 66% were doing manual testing of their data sources
  – 49% had automated checks on external data
  – 45% were doing automated regression testing of data sources
  – 36% were doing data security/privacy testing

Source: SA+A 2018 Data Quality Survey

© Scott Ambler + Associates
Current State: Data Quality

93% of respondents indicated that their organization had data quality problems.

Their strategy to fix the problems:

- Database refactoring: 44%
- No strategy to fix: 27%
- Rewrite all applications at once: 0%
- Hope we don’t make it worse: 29%

Source: SA+A 2018 Data Quality Survey
Why Do We Have Data Quality Challenges?
Scale of -10 (highly disagree) to +10 (highly agree)

- Data professionals adept at modern techniques: -0.8
- Developers adept at evolving data sources: 1.5
- Data professionals find it easy to work with developers: 2.6
- Management supports fixing legacy data sources: 2.6
- Developers find data professionals easy to work with: 2.9
- Dev teams willing to use legacy data: 5.3

Source: 2018 Data Quality Survey

© Scott Ambler + Associates
Strategies for Database DevOps

- Vertical Slicing
- Clean Architecture and Design
- Agile Data Modeling
- Database Refactoring
- Database Regression Testing
- Continuous Database Integration
- Configuration Management
- Operational Database Monitoring
Vertical Slices of a Solution

• Every iteration a disciplined agile team produces a working solution
• Functionality is added in “vertical slices”

• Example slicing strategies for DW/BI:
  – One new data element from a single data source
  – One new data element from several sources
  – A change to an existing report
  – A new report
  – A new reporting view
  – A new data mart table
Considerations for Clean Data Architecture

- **Consistency**
  - ACID (Atomic Consistent Isolated Durable) transactions
  - Eventually consistent
  - Good enough
- **Problem/Solution fit**
  - SQL vs. Hierarchical vs. No-SQL vs. files
- **Latency**
  - How fast does access need to be?
- **Security**
  - Who should access which data?
- **Scalability**
  - Do you need to cluster?
  - Support multiple physical locations globally?
- **Sourcing**
  - Get the right data from the best sources
- **Auditability**
  - Ensure that you know where and when data came from
- **Historical storage**
  - Retain previous versions of appropriate data
Strategies for Clean Database Design

- Normalization
  - Are you storing data in one and one only place?
- Design for the database type
  - OLTP (transactional ➔ normalized design)
  - OLAP (analytical ➔ denormalized design)
- Tables and columns should be cohesive
- Future proofing?
  - Maintaining historical data values
  - Soft vs. hard deletes
  - Truly unique surrogate keys
  - Implement all relations as many-to-many
- Set and follow naming conventions
WE NEED DETAILED DATA MODELS FROM THE START OF THE PROJECT.

THAT'S TOO RISKY! WE NEED TO WORK IN A MORE AGILE MANNER.
Agile Data Modeling: Concept

- Data modeling is the act of exploring data-oriented structures
- Evolutionary data modeling is data modeling performed in an iterative and incremental manner
- Agile data modeling is evolutionary data modeling done in a collaborative manner

There is nothing special about data modeling!
Agile Data Modeling: In Practice

- Inception: Initial Modeling
  - High-level conceptual modeling
- Construction
  -jit Modeling
    - Detailed physical data modeling (to generate DDL)
  - Test-Driven Development
    - Detailed specification
WHAT IF I TOLD YOU

THAT IT IS TRIVIAL TO EVOLVE A PRODUCTION DATABASE SCHEMA?
Why Evolving Databases is Thought to be Hard

Production databases are often highly coupled to other systems, services, data sources, …
Database Refactoring

- A database refactoring is a simple change to a database schema that improves its design while retaining both its **behavioral and informational semantics**

- A database schema includes structural aspects such as table and view definitions; functional aspects such as stored procedures and triggers; and informational aspects such as the data itself.
The Process of Database Refactoring

Verify that a refactoring is needed

[Not Needed]

Choose the Right Refactoring

[Pass]

Run the tests

[Fail]

Write a Unit Test

Change your schema

Migrate Data (optional)

Update External Access Programs

[Fail]

Run the tests

[Work continues]

Announce The Refactoring

Version Control Your Work

Deprecate the Original Schema (optional)

© Disciplined Agile Consortium  @scottwambler
SO YOU'RE TELLING ME THAT DATA IS A CORPORATE ASSET

YET YOU DON'T HAVE REGRESSION TEST SUITES FOR YOUR DATA SOURCES?
Agilists develop regression unit and acceptance test suites for their applications, why not for databases too?
If database testing is done at all these days, it is at the black-box level. This is a good start but often isn't enough.
You must be able to put the database into a known state, therefore you need test data (generation)
Test/Behavior Driven Database Development (TDD/BDD)

- An evolutionary approach to database design
- Driven by tests, not models
- A practical approach to ensuring database quality
- Completely different than traditional approaches to database design
- Really just one part of BDD, not a stand-alone activity

- May/June 2007 issue of IEEE Software
Continuous Integration and Databases

- Part of building the system is building the database (if it changed)

- Challenge: Tests SHOULD put the database back into a known state, but sometimes don’t
  - You will want to rebuild the (non-production) database from scratch every so often

- Challenge: Database accesses take time
  - Some test suites will test against DB mocks
  - You still need to test the actual database occasionally
I PITY THE FOOL

WHO DOESN'T HAVE CONTINUOUS INTEGRATION (CI) IN PLACE.
Configuration Management

- All work products should be stored in a versioned repository
- Maintains the integrity of the system and all supporting work products as it evolves
- Facilitates change in a controlled fashion
- All revisions kept and who made what changes to all work products
- Potential benefits include the ability to:
  - Manage versions of systems across environments
  - Rollback to previous versions
  - Modify work products in parallel and then merge
  - Find source of defects injected
Operational Database Monitoring

Potential operational intelligence about operational data sources includes (but is not limited to):

- Availability
- Buffer/cache size
- Usage trends
- Read load
- Write load
- Data source size
- Data source connection time
- User connections
- User-data access (security access rights)
- Privileged user (super user) access
- Application
Exercise: Explore a Technique

• On your table is a sheet describing one of the key Database DevOps strategies.

• Within your group, discuss the following issues:
  – Advantages of the strategy
  – Disadvantages of the strategy
  – How to gain skills in the strategy (be specific)
  – Tooling support

• For each issue
  – Capture up to three ideas/items about it
  – Use one sticky note per idea
Parting Thoughts
Data Must Become a True Corporate Asset
Disciplined DevOps

DisciplinedAgileDelivery.com/disciplineddevops/

© Disciplined Agile Consortium
Thank You!

scott [at] scottambler.com
Twitter: @scottwambler

AgileData.org
AgileModeling.com
DisciplinedAgileConsortium.org
DisciplinedAgileDelivery.com
Would You Like This Workshop in Your Organization?

Contact us at ScottAmbler.com
Scott Ambler + Associates is the thought leader behind the Disciplined Agile (DA) framework and its application. We are a boutique consulting firm that advises organizations to be more effective applying disciplined agile and lean processes within the context of your business.

Our website is ScottAmbler.com

We can help