The "Agile" Approach To Database Schema Design

This paper is an adjunct to our weekly webinar series about Database Design. The "Agile" Approach To Database Schema Design is a topic that garnered a lot of attention.

The Agile Manifesto

The Agile methodology was created to help design better IT programs. It is based on the Agile Manifesto that is comprised of 12 principles. We will concentrate on the following important points:

- Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
- Business people and developers must work together daily throughout the project.

We summarize how this process works, in the following diagram:
An important item is the collaboration of the end users with the developers. In addition, if you examine the above 4 steps, you can see that it is basically about:

1. Building **better applications**
2. Building them **faster**

What does this really mean?

1. Building better applications delivers:
   a. Pristine Information
   b. Great Performance
2. Building them faster delivers:
   a. Reduced Cost
   b. Short Delivery Time

**The "Agile" Approach To Database Schema Design**

We believe we can have a similar approach to designing better database schemas.
It is basically about:

1. Building **better schemas**
2. Building them **faster**

What does this really mean?

1. Building better schemas delivers:
   a. Pristine Information
   b. Great Performance
2. Building them faster delivers:
   a. Reduced Cost
   b. Short Delivery Time

**Who will be responsible for the feedback?**

We can see that it is very difficult to decide at this moment which agent will be responsible to provide the feedback to the database architects; It is not the programmers responsibility to deal with database schemas and users probably have never heard about database schemas. In a moment we will show how the feedback will be provided.

**Consequences of a poorly designed schema**

If your schemas are flawed you will have:

- High development cost
- Late delivery of applications
- Poor performance
- Poor quality of information

The first 2 items occur because of the way we currently develop IT programs and the last 2 items occur because of the poor quality of the database schemas.
The current development (& maintenance process) is a non-Agile process; let us review it.

The Current Development Process (non-Agile)

When debugging the programs, if the developers find errors that are not program errors, they will send this information to the database architects that have to make changes to the schema and then send it back to the developers that must continue testing the programs. **This going back and forth is the main reason why applications are delivered late and the cost is much higher.** The same happens with the maintenance.

The "Agile" Development Process

We propose the following "Agile" Development Process:

Figure 4 - The Proposed "Agile" Development Process
The "Agile" Maintenance Process

We propose the following "Agile" Maintenance Process:

![Diagram of the proposed "Agile" Maintenance Process]

Figure 5 - The Proposed "Agile" Maintenance Process

So, if we use the "Agile" development & maintenance processes, we will reduce cost and shorten delivery time of the applications because we are checking and fixing the errors very early in the process.

**Reason why we produce poor information and poor performance**

Joe Celko, a top database expert and published author, states in his paper *It's the Code, Stupid!*

“A poorly designed database schema affects negatively:

a) the quality of the information and

b) the performance of the application”

Because of the above reasons, we see that flawed schemas are responsible for major issues.
Introducing Database Examiner

Database Examiner is a schema validation tool, based on 2 knowledge bases:

1. The Relational Theory
2. The Performance Best Practices

It provides 60+ diagnostics, 80% of which are integrity related (quality of information) and 20% are performance related.

Database Examiner is unique in the market place. No other solution provides such a set of comprehensive diagnostics. Examiner processes ERwin data models, SQL/DDL scripts and RDBMS schemas from DB2, ORACLE, SQL Server and Sybase. It has a powerful TeachMe facility and it is extremely fast. Processing a schema with a couple thousand tables takes a few minutes.

As of today, there are 2,500+ organizations that use Database Examiner.

Who provides the feedback?

Getting back to the "Agile" Approach To Database Schema Design we can show that Database Examiner will provide the feedback to the Database Architects.

![Diagram]

Figure 6 - The Final "Agile" Approach To Database Schema Design

Conclusions

By using Database Examiner and the proposed "Agile" Development & Maintenance processes, your organization will produce much better applications (pristine information & great performance, with reduced cost and shortened application delivery time).
Applying the "Agile" Approach To Database Schema Design

1. **Divide the project into manageable sub-projects**

   ![Diagram of dividing projects into sub-projects]

   Figure 7 - Dividing the projects into sub-projects

2. **Develop each sub-project, validate & fix it**

   ![Diagram of developing and validating sub-projects]

   Figure 8 - Developing each sub-project, validating & fixing each one

At this point, the developers could start coding the programs for each of the sub-projects.
3. **Compare 2 validated sub-projects, solve the discrepancies, merge them, validate the new combined sub-project, and fix the new validated sub-project**

![Diagram of project management process]

**Figure 9** - Merging sub-projects, validating & fixing the combined sub-project

**Note**: When merging sub-projects, we must insert the relationships between them.

4. **Use the same procedure above until all the sub-projects have been merged, validate and fixed.**

![Diagram of project management process]

**Figure 10** - The Final project after combining all the validated sub-projects
About DBE Software, Inc.
DBE Software, Inc. is in the information quality business. To date, over 2,500 clients worldwide have successfully deployed our technology.

6842 Elm Street
McLean, VA 22182 - USA
+1-703-847-9500
www.dbesoftware.com
info@dbesoftware.com

Prepared by:
Luiz C. Siqueira
Founder & CTO
LCS@dbesoftware.com