EMBARCADERO TECHNOLOGIES®

White Paper

A New Tooling Charter for Data-Driven Software Development

Embarcadero Technologies, Inc.

September 2009

Corporate Headquarters
100 California Street, 12th Floor
San Francisco, California 94111

EMEA Headquarters
York House
18 York Road
Maidenhead, Berkshire
SL6 1SF, United Kingdom

Asia-Pacific Headquarters
L7. 313 La Trobe Street
Melbourne VIC 3000
Australia

SUMMARY

Businesses today are gathering and storing data at a breakneck and ever-accelerating pace, feeding the unquenchable imperative to leverage information for growth and transformation. Enterprise data and the demand for data-driven applications have grown faster than the IT organizations that support them. The skills & disciplines required to transmute data into business opportunity are varied and complex and have likewise failed to keep pace with the growth.

DBAs who have historically played an important role in the management of data have steadily become less and less involved in data-oriented activities, and instead spend most of their time managing database infrastructure. This has been a natural side-effect of the explosion in data. DBAs are very highly leveraged resources, and their role has become focused on the strict essentials: keeping the databases up and ensuring recoverability, security, and performance.

The increased need for data-related skills has led to a growing gap, which developers are often times called to fill. Indeed, developers are the ones who are the most realistically and pragmatically suited for stepping up their responsibilities and bridge this data management gap, as they are the closest to the code and to the application and business requirements. The average developer however, does not typically undergo the same rigorous data training as a DBA, data modeler, or data architect, which normally cover topics such as relational theory, structured query language (SQL), database structures, and data movement and transformation to a great level of depth.

If indeed developers are assuming a greater responsibility for data, then it is important for organizations to understand what developers need in order to be effective in this new role. An important facet of this question pertains to communication between the DBA and developer, including the ability for the DBA to gain ready visibility into what the developers are working on, so that they can intervene at the right times to provide guidance.

This paper explores this shift and the notion that the data gap can be bridged to a great degree by improvements in communication. It further proposes that data and database tools can not only be a great productivity aid, and play a role in assuring quality, but the tools themselves can become an important vehicle for normalizing communication, also reducing the need for an intensive and ongoing training regimen. By providing developers with the tools they need to carry out their data-related duties, the business can realize substantial and enduring gains in the form of project quality, time to market, productivity, and job satisfaction.

Tools can help organizations bridge the data gap in three ways: by growing developers' capacity and self sufficiently with data; by helping DBAs to become more effective in their own roles—freeing them up to re-assume some of their more traditional data management roles; and perhaps most importantly, by serving as a glue to bring the DBA and developer into closer sync with one another, facilitating workflow across projects. Data and database tools can serve as a lingua franca that ensures clear and unambiguous communication amongst developers and DBAs.

THE PROBLEM OF DATA

Tightened budgetary controls have opened businesses and IT organizations' eyes to the increased costs and waste resulting from poor data management and handling. A study carried out by Experian QAS titled "The hidden costs of Poor Data Management" found that companies across the globe were losing on average 6% of their revenue due to poorly managed data. The same study found U.S. companies being hit the worst with losses of 7.3% of revenue. Not only does this hinder financial flexibility, but also the ability to operate effectively and productively.

As IT organizations and businesses try to cope with budgets, efficiency, and productivity, they must do it while suffering under the weight of poorly structured data architectures and databases. And it doesn't stop there. Poorly written and inefficient application code underpinning data-driven applications further compounds the problem, while poorly coded and structured data has a direct effect on environments where the experience of the interaction is a lucrative commodity. Thus, applications that breakdown or run slowly because of badly coded and inefficient queries place an unacceptable level of burden on end-users as they struggle to execute their tasks.

Badly structured data models raise costs as additional work is required to cleanse the data, improve access to it and allow others to better interpret it for subsequent reuse. The complexity that can arise for providing additional layers to improve both the quality of and access to the stored data, limits the value of any productivity features provided and places additional stress on the processing facilities.

ROLE OF THE DBA: DATA MANAGEMENT VS. DATABASE MANAGEMENT

The DBA role has evolved greatly over the last couple decades. In this time, the sophistication of the database technology, as well that of the underlying systems, together with data volumes, has grown dramatically. This, together with the growth in the number of databases that each DBA must manage, and the greatly increased size of each database, has changed the way that DBAs must work.

Previously it was common for DBAs to become highly involved with the application requirements and design, and with the meaning of the data. They were often deeply involved not merely in the modeling and architecture of the system, but played a hands-on role in activities such as data transformation, validation, and movement, in addition of course to driving de-normalization, indexing, partitioning strategies, and the like. DBAs came to develop a very deep understanding of the data and its meaning, and were often called upon to develop and carry out complex queries and to work directly with the data in various ways.

SHIFTING RESPONSIBILITIES: DEVELOPERS AND DATA

In the past, when developers needed to access to the data within an enterprise database, it was not unusual for a DBA to review each and every SQL statement and stored procedure. As DBAs assumed responsibility for more and more systems, and as developers and development projects came on at an ever increasing rate, DBAs started to no longer have the luxury of being so deeply involved in the activities surrounding access to data. This created a natural shift

whereby developers began to gradually assume more responsibility than before with regard to data- and database-oriented activities. A phenomenon that occurred in correlation with this shift was the proliferation of data-driven applications, which allowed applications to become increasingly flexible, and required developers to interface more often with databases. It is possible that the move towards data-driven applications contributed to the shift. However it is more likely the reverse: that the applications became data driven precisely because developers were gaining greater facility and control over database logic.

Communication and the Developer-DBA Dynamic

As developers take on a more important role with data management and DBAs become more and more system and architecture oriented, there is a risk that these two important groups might start to lose touch with one another. Information systems play a more important role than ever before, and it is more important than ever that lines of communication between DBAs and developers, as well as with data modelers and architects, be clear and strong. The many opportunities for disconnect can be averted after one understands the pitfalls:

1. The natural forces that have caused DBAs to become infrastructure focused can also make them less accessible to developers, increasing project risk. DBAs play a key role in reviewing code and database changes, helping to catch problems early in the project lifecycle, before they can cause too many problems;
2. Lack of communication and documentation of existing data assets can result in lack of reuse, leading to data redundancy and problems with data quality;
3. Ambiguous ownership of the data can lead to various problems, affecting everything from help desk escalation (who should be responsible for fixing the problem) to more persistent data quality issues that fail to get fixed because there is no clear point of accountability;
4. Geographical and organizational boundaries, including differences in language and work culture, can make it more difficult to communicate, and lead to an increased potential for misunderstandings.

The key to solving all of these issues resides in improved communication. Many methods have evolved for improving communication on projects: regular meetings, cultural sensitivity training, and clear process and role definition. These "soft" factors are important, but they alone cannot be successful. What needs to be communicated when working with data is technically rich and complex. Consider issues such as structure, sensitivity, ownership, quality, accuracy, and performance characteristics. It is not enough to merely converse about these topics. Rather they must be clearly and precisely documented to a great level of depth. This is a requirement of any handoff, whether between individuals, between projects, or between phases of a project. This is where tools can come in. Software tools specializing in managing data and databases can, when used by developers, DBAs, and data architects, serve as a lingua franca to facilitate and normalize project communication. Tools can serve as a key mitigation factor for plugging the data gap, because they can not only document and report on complex systems, but they can also help uncover patterns and issues that would otherwise be very difficult to detect.

A New Order for Tools

It is interesting to think of tools as facilitators for communication. This is a new way of thinking, as most tools are usually thought of as productivity aids. However this view is consistent with

contemporaneous usage patterns. Data modeling tools for example, have come a long way from their initial state. Today they are used not only to design databases, but to document, visualize, classify, tag, centralize, and report on data assets. Today's data modeling tools can answer questions about what is stored where inside of an enterprise, or what subject areas, tables, or even columns contain sensitive or classified information. They are also beginning to blur the line with what has traditionally been the development world, and are now used to model data in transit, generating XML schema as in addition to just database schema.

Another good example is database change management tooling. Such tools can capture information about specific state of a database at various points in time, and report on how the database has evolved: structures, permissions, settings, stored procedures, and the like. Features such as DDL generation greatly relieve the pressure on both developers and DBAs.

SQL tuning is a third example of how the same tools in the hands of a developer and a DBA can help make communications more efficient. As mentioned previously, today's DBA no longer has time to review each and every query. However developers are generally too busy producing to acquaint themselves with every last nuance of efficient SQL writing… particularly since every DBMS (and version thereof) brings a variety of different options and syntax variations. Providing a SQL tuning tool to developers lets developers handle the majority cases, so that DBAs can spend their time focusing on the more difficult problems. The end result is that developers will make the most effective use of the DBAs time, and their own.

The modern data and database tool set has something to offer both developers and DBAs. For both groups to standardize on a common set of tools provides a universal language, which allows them to design, build, and run information systems smoothly and consistently.

MAKING THE RIGHT TOOL SELECTION

The database tools market goes back nearly twenty years, and today, a variety of vendors offer solutions to various point problems: generalized database management, performance, database change management, security & compliance, and so on. If tools can be a glue to help bring DBAs and developers more closely into sync, it is important to understand what options are available, and that the options cover both the needs of the DBA and the developer.

Database vendor tools for example have a history of catering to the DBA, and less so to the developer; and few would disagree that for many years they were a secondary focus compared with the core database business. The quality of the database vendors' tools has steadily improved over time. However to the degree that tools cater to only one group, the invaluable opportunity for tools to become a facilitator of communication is lost. Another limitation is that the each database vendor has its own set of native tools, each with its own interface and learning curve. Independent vendors who are DBA and developer focused, and who offer multi-platform support, are often a better choice for large organizations, particularly those that rely on more than one database platform.

An important trend in modeling tools (especially data modeling)has been to centralize the storage and management of models, treating them as the valuable enterprise asset that they are, but also allow collaboration (with proper access control) as well as centralized reporting and model metadata analysis. A modern toolset will certainly provide such options for centralization, as well as breadth of functional coverage (ideally spanning multiple roles). Developer tools should support centralization via Source Code Control, so that database

artifacts can be managed alongside application artifacts, directly from within the tools. Specialized solutions for database change management should be used alongside source code control, for management of database objects. Software Source Code Control is limited in that it does not understand database syntax, and is not able to connect to a database, nor generate a synchronization script that could be run inside the database. In the same way, software source control is not adequate for managing data models, because it can only manage files, and cannot decode or represent models, sub-models, and objects within each model.

Databases are highly complex, and evolve rapidly. It is important that the tools which one chooses support the latest versions, but also the older versions that are almost always found in enterprises. A good tool will understand the syntactic and feature differences between, for example, Oracle 9i, 10G, and 11G; or SQL Server 2000, 2005, and 2008, and make use of those features as the need arises. A tools vendor should have a strong track record, and provide a broad-based solution that is mature and robust. Multi-vendor solutions are certainly to be found, but there are many benefits to consolidating around a single vendor whose solution stack meets most or all of one's requirements. Single-vendor solutions provide economies of scale, and are generally easier to purchase.

The challenge of making the right tool selection is defined by two driving forces: ease of access to the right tools, and the richness in the functional capability of the tool to productively and efficiently support the right roles. In essence this translates to a tooling and licensing model that allows access to the right tools at the right time without a great deal of administrative overhead.

The Embarcadero Value Proposition

With a 15-year track record of helping many of the world's largest companies run their databases and design and build database-oriented applications, the strategy at Embarcadero is to provide the best tools and services that support architects, developers, and DBAs, leveraging the synergies across their roles.

The acquisition of CodeGear™ from Borland®, Inc. in 2008 added a rich set of highly regarded and widely adopted developer focused tools and services to our strength with a broadening portfolio of developer solutions with independent database development and management tools. Embarcadero's product portfolio focuses on making the lives of those that design, build, and run software systems more productive, effective, and efficient. The tools provided support the lifecycles and technologies of all the major database platforms and development models (e.g. JAVA™, .NET, Oracle®, SQL Server, DB2®, MySQL®, and more), providing clients with the flexibility of choice and portability they require.

Embarcadero's long and combined history in working and supporting the development and database community has led to an understanding of the challenges that exist today for software development and delivery teams. With a deeper and broader tools portfolio, the gap between DBAs and developers can be closed by providing tools that address both communities effectively. In recognizing the management overhead of tool selection amid licensing, pricing, and configuration challenges, Embarcadero has developed a new tooling framework and licensing model that will unleash them from their constraints.

ALL-ACCESS: THE NEW TOOLING CHARTER AND PASSPORT TO MINIMIZING TOOLING BUDGET AND MAXIMIZING YOUR TOOLING PORTFOLIO

Launched by Embarcadero in February 2009, All-Access is a new approach to buying, deploying, and using tools. It greatly simplifies buying decisions by providing tools via a "tool chest" model. Rather than buying standalone tools a-la-carte, customers buy an All-Access Pass and subscribe to a yearly membership to gain access to perpetual licenses within Embarcadero's solution stack. This includes full support and a unique on-demand deployment model that greatly speeds the acquisition and deployment of tools. All-Access is highly cost-effective, costing roughly the same as it would cost to buy two to three products.

Embarcadero® All-Access™ provides the following:

- A licensing model that gives access to a range of related and pre-integrated development and database tools through a discounted pricing model;
- On-demand click-and-run deployment framework underpinned by technology that allows instant access to fully featured tools with no prior installation;
- Membership based support and maintenance program which, unlike a subscription, offers perpetual licenses, so that customers own the software from day one.

License subscription models are not new. However, the All-Access model is fundamentally new in a number of ways and addresses many of the concerns that customers have raised over the years related to acquiring, installing, and optimizing usage of software, as well as managing and tracking licenses. All-Access uses a role-flexible model so that an All-Access pass can be floated from a DBA and back to a developer. Because DBAs and developers are using the same tools, consistent and smooth communication is gained, together with the productivity and reliability gains that come from using best-in-class tools.

EMPOWERING DEVELOPERS AND DATABASE PROFESSIONALS ALIKE

Because it functions as a tool chest, All-Access takes on a different shape for the DBA than it does for the architect, or for the developer. Each individual can pull the tool he or she needs, when it is needed. At different phases in the project, one's tooling needs may vary; and different projects call for different technologies. All-Access is the perfect solution, because like a mechanic's chest of tools, it gives users access to the right tool at the right time.

Here is an example scenario. On Monday a DBA may use the tools inside All-Access to create some new users (using DBArtisan®, Embarcadero's award-winning database management tool) and to profile a database using DB Optimizer. On Tuesday, he or she could use Embarcadero® Change Manager™ to compare a test and production database and automatically generate a sync script to carry the relevant changes forward into production. On Wednesday, he or she can reverse engineer a new database using ER/Studio® Data Modeler. And all the while Performance Center has been monitoring the database and sending alerts whenever a problem is detected.

By the same token, the same All-Access Pass for a developer provides a powerful set of tools to develop software and database solutions, any of which can be used (when the time is right) to solve a particular problem. These include:

- Rapid application development tools for native Windows development and .NET
- Java™ development and optimization tools
- Design and architecture tools to support data model browsing, data modelling, and process modelling
- A SQL IDE for database development, along with SQL tuning facilities
- Web development tools
- Embedded and workgroup databases

All-Access not only helps each group carry out their own duties, but can help improve the working partnership between the database engineering and management teams and developers by providing a common language and capability.

Affordable and Flexible Pricing and Licensing

There are many other tools inside All-Access such as DBArtisan®, Rapid SQL®, Delphi®, JBuilder®, C++Builder®, the InterBase® SMP database and many more. All these tools can improve the way that developers and DBAs work both individually and as part of a team.

All-Access is designed for those IT organizations looking for more cost-effective and affordable pricing models as budgets tighten, and team members are called upon to do more. The cost of each level of All-Access has been set at the combined cost of just a few tools in each level. This means that any tools beyond that point are effectively free. By getting more tools than you pay for, you can begin to implement solutions that would previously have been too expensive.

All-Access Bridges the Gap

Embarcadero All-Access is the ideal tool set for supporting this new tooling charter for data-driven application development, and to break down the barriers for greater team collaboration, communication, and productivity. Please visit http://www.embarcadero.com/products/all-access to learn more and to request a trial.

Embarcadero Technologies, Inc. is a leading provider of award-winning tools for application developers and database professionals so they can design systems right, build them faster and run them better, regardless of their platform or programming language. Ninety of the Fortune 100 and an active community of more than three million users worldwide rely on Embarcadero products to increase productivity, reduce costs, simplify change management and compliance and accelerate innovation. The company's flagship tools include: Embarcadero® Change Manager™, Embarcadero® RAD Studio, DBArtisan®, Delphi®, ER/Studio®, JBuilder® and Rapid SQL®. Founded in 1993, Embarcadero is headquartered in San Francisco, with offices located around the world. Embarcadero is online at www.embarcadero.com..