InfoAdvisors

7 Steps Every Successful Data Architect Follows to Build Stronger Teams

Karen Lopez, InfoAdvisors

InfoAdvisors

Karen Lopez
@datachick

www.infoadvisors.com

Helping You Add Business Value to Your Information Management Resources

#TEAMDATA

An updated version of *Seven Steps Every Successful Data Architect Follows to Build Stronger Teams*, 2014

This white paper is sponsored by IDERA Inc.
ER/Studio is a registered trademark of Embarcadero Technologies, Inc., a wholly-owned subsidiary of IDERA Inc.

When teams work together with a common goal and a passion for success, their projects succeed. When the opposite occurs, everyone loses, including the team, the employer and their customers.

In the course of my normal consulting assignments, I'm often told that there's something wrong with the data models being prepared by the team architects, but what I often find is that there is a significant disconnect between what the architects are doing and what the rest of the team expects them to do.

In this paper, I will look at some of the problems that project architects face in working with other IT roles and seven easy and inexpensive steps they can take to ensure a collaborative team environment.

Easier Projects, a Better Product and a Stronger Team

Is your team working with you or against you?

Most IT professionals have experienced contention and conflict on at least one of their projects, often due to misunderstandings of what data architects are responsible for delivering. This mismatch of expectations can slow down a project and lead to weak products. In some cases, organizations can get to the point where more effort is spent on resolving contentious issues than on getting real work completed.

Collaborative Tasks Are Completed Faster

In my experience, when teams are working toward the same goal, both individual and collaborative tasks are completed faster. Less time and effort are spent revisiting decisions, debating courses of action, and critiquing end results.

Collaborative Tasks Are Easier

When there is less contention and more trust among team members, tasks are easier to complete because there are fewer distractions. Team members aren't escalating decisions to their managers for resolution.

Collaboration Increases Confidence in IT Teams

When business users see IT professionals actively and loudly debating the merits of some technical decision, they lose confidence in IT as a whole. When they continue to see the same issue raised throughout a project or carrying over to other projects, they lose confidence in the IT organization, even if only a few are the cause of the issue.

Jerry Weinberg writes:

"If you use the same recipe, you get the same bread".

If you want a more collaborative team, you have to change how and where you work. This paper includes seven easy and inexpensive steps to improving your ability to collaborate with the technical members of your teams.

The 7 Steps

Easy and Inexpensive Steps to Better Collaboration

Now that we've covered the benefits of better collaboration, let's look at the steps we should take. You may already practice a number of these, but I find that using all of them can repair a dysfunctional team relationship.

Step 1: Get out of the Ivory Tower, Physically (Location, Location, Location)
Many organizations choose to locate staff with their managers, leading to staff sitting next to people who have the same function, but not with the teams they support.

Data architects should be physically located where the teams they support work so that they are instantly available to answer questions and work through issues. This is the number one change I recommend to architects when they tell me that they have trouble working with developers or database administrators (DBAs). Yes, we do have telephone, instant messengers, and web meetings, but nothing beats just being there when people are scratching their heads, wondering what the heck a *RETAIL TRANSACTION LINE ITEM MODIFIER EVENT STATUS CODE* is. In fact, most will make an assumption and carry on with their next step if they have to stop and find someone to ask…and I don't blame them.

Being physically located with development staff is the number one change I recommend to architects having problems working with teams.

Nothing builds trust better than actual face-to-face interaction with all participants involved in a project.

Managerial Control

Many architects find themselves reporting to a manager who insists that her group sit together, close to her desk, so that she can properly manage their daily activities and track progress. In those cases, it is important to build trust and visibility into your daily activities while continuing to reinforce the fact that your job is very iterative and requires an incredibly collaborative relationship with the development teams with whom you work.

If your manager still insists that architects have desks together in the same location, you might want to ensure that your developers and database administrators have access to comfortable visitors' chairs, worktables and whiteboards.

Multi-location and other location projects

If you are a data architect who supports multiple projects or projects in other locations, then you need to be more creative with how you can "be there with them". Online meetings and teleconferences are somewhat effective, but it is difficult to gauge responses of people on the other end of the line – Are they working on an e-mail as they listen? Can they hear what you are saying? Do they understand what you are describing?

The advent of video for attendees in virtual meeting services has helped a great deal. You should ensure that your team has access to video cameras or virtual presence technologies.

Where development and architecture activities take place in different locations, I recommend that an architect travel to work with the project developers and database administrators during the initial and final development efforts, even if there are significant travel costs. I have seen too many misunderstandings and defects, costing projects hundreds of thousands of dollars, largely due to having architectural roles physically separated from development staff. Sometimes a single architect can provide process, data, and technical architecture support to save costs, but in the end, skimping on travel costs usually leads to much higher fix and repair costs later in the project.

Sometimes a single architect can provide process, data, and technical architecture support to save costs, but in the end, scrimping on travel costs usually leads to much higher fix and repair costs later in the project.

Let's Do Lunch – or a Coffee

If you can't sit with your team members every day, then minimally you should be lunching with them or going to company events with them – anything you can do to let them understand that you are indeed on the same team and that any professional debates are just that, nothing personal.

If you are having a more acute problem with collaboration, I highly recommend breaking into smaller groups, even one-on-one situations, to talk about why collaboration is not going well. One drink doesn't make for a team, but addressing the problem head on is the best way to get better understanding and to increase trust.

Step 2: Get out of the Ivory Tower – Logically

Most new developers are part of the younger generation since development is an entry-level position in most companies. Architects, on the other hand, tend to be more experienced, having worked their way from developer or database administrator to architect. In fact, most data architects have at least 10 years of IT experience. This difference in experience is enough to cause significant problems working across generations.

If you are managing or supporting the new generation of IT worker the same way you want to be managed, you might be doing the opposite of what they'd like you to do.

Use Modern Communication Tools

In general, your recently hired workers tend to strive for a greater life-work balance. They are willing to put in a great number of hours, but would rather have more time off than be vested in a company savings plan. They want to use many social media and online features such as wikis, blogs, online voting, and instant messaging. Traditional top down communication styles and e-mail are perceived as outdated methods. They are frustrated by static, document-driven requirements processes that result in deliverables that are hard to update, comment on, or take away with them. I'm frustrated with that, too.

Managing the new generation of workers requires new approaches and new communication styles.

Generate and Maintain Modern Artifacts

Team members are no longer interested in reading a 175-page document on how a data model works. They insist upon having an interactive, up-to-date, searchable, shareable and comment-driven set of tools for working with models and other project deliverables.

You may want to deploy newer collaboration technologies that encourage team members to collaborate via many options and platforms. That includes mobile devices.

Adapt To the Communication Style the Rest of the Team Uses

If you‘ve worked long enough that you understand the bureaucracy and hierarchical communications structure, you're probably experienced enough to change when, where, and how you communicate with the rest of your team. You must adapt your style and your deliverables to work with a newer generation of IT professionals.

Step 3: Know Their Pain

If you can't list the top three complaints your team members have about your models, you haven't been listening closely enough. That also means that your team members know you haven’t been listening.

Nothing sets you up for constant friction with team members more than ignoring their pain. Certainly there are times when you can’t satisfy their needs, but often the solution is easily at hand. For instance, many inexperienced developers are puzzled by generalizations in a data model. Why have something called GEOGRAPHIC ZONE when something like STATE and CITY would work just as well? Why have a STATE table when they can just put all that information in their application code? ***You*** know why you've used a generalization, but ***they*** might not.

If you can't list the top three complaints your team members have about your models, you haven't been listening closely enough.

Create Data Walkthroughs

You can help in these cases by preparing data walkthroughs and use cases that specifically show the benefits of using a particular approach and the costs of using a more project-specific solution.

Prototypes that can be generated directly from your models are great tools for demonstrating how a model works. Even generating a spreadsheet from ER/Studio with one tab per table can help you show sample data. I use this technique for the most generalized versions of data models.

Demonstrate, Don’t Just Tell

Another common complaint is that the data model "takes too long". Sometimes that is a valid complaint, but often it reflects how much the team does not know about the problems being addressed by the application. One technique that works is to log issues to be addressed in the issue management system. This allows management and developers to see that understanding the issue, not just modeling, is what is taking time. It demonstrates the exact obstacles and the time it takes to negotiate a resolution to a complex question.

If you allow developers and database administrators to have the impression that architects don't understand their needs, they will have little motivation to understand yours.

These examples are provided to demonstrate that perceptions must be managed. If you allow developers and database administrators to have the impression that architects don't understand their needs, they will have little motivation to understand yours.

Step 4: Treat Models As Well As Other Deliverables.

Models are products and require the same level of configuration, issue and problem management that application code does. This means that models should have versions that are separate from the code; they should be formally released; and should have test plans and data that focus on the completeness and accuracy of the model separate from the application test cases.

Team members should be able to log defects for the models. Architects should be creating test data or at least sample data for their models, especially those much generalized structures that will provide future benefits.

Architects should also be able to trace changes to the model back to a change request. These change requests submitted during development should be collectively reviewed and prioritized by the architects, development team and database administrators.

If you aren't formally versioning and releasing the models, you are sending a clear message that models are not important enough to do so.

If you aren't formally versioning and releasing the models, you are sending a clear message that models are not important enough to do so.

Step 5: Ensure No Models Get Thrown "Over the Wall"

Some IT professionals see data models as deliverables that are completed early in the project, and then handed off to others to decide how and when they use the models.

This detached approach is almost always guaranteed to cause a breakdown in communications between the business needs and what gets implemented. A modeler should be involved in how the requirements are captured as well as how they are implemented. She should understand where they are implemented, and what changes, if any, have been made due to technical constraints or performance requirements. Data modelers are the mediators between business requirements and physical implementations.

Monitor and Trace

Architects have a responsibility not only to prepare the architecture, but also to ensure that what is built satisfies the goals of the original architecture. It is their responsibility to provide support for those implementing the models and to measure their compliance with the models. Deviations can be accepted, but architects who build a design and move on to other projects are skipping out on the most important tasks on a project – implementation.

A "throwing over the wall" approach will lessen the value of data models – and increase costs.

So how do architects ensure that models are used? They ensure that they are part of the implementation process. They ensure that there is adequate support during the development process to meet the needs of the team. They work with project managers to develop more iterative methods and schedules. They inspect the work of the developers and database administrators and raise issues with architectural compliance. They measure and report on how well their designs worked in the real world. They measure and report defects for their own products. In short, they treat their models and designs as products, not just inputs into someone else's product.

Step 6: Model the Business, Not a System

The most critical mistake you can make is treating models as if you personally own them. The models should be presented as belonging to the business and stewarded by the modelers. That means sharing them openly, providing access to those who want it, keeping extra printouts available, offering training on how to read them and making every effort to make them clear and understandable.

Models and their underlying metadata are corporate assets to be managed by a partnership of modelers and the business.

Models and their underlying metadata are corporate assets to be managed by a partnership of architects and the business. In fact, we call all the data and process modeling we do *Business Models,* to demonstrate that these models are not just temporary inputs into the development process of a single project but also longer term corporate assets. We create *business logical models* and *business physical models*.

Keeping the emphasis on the business side of things reinforces the fact that we architects want to solve real problems, not just technical curiosities.

Step 7: Stop Bad Mouthing Them

Architecture and development are naturally in conflict with each other. A good architecture takes into account legacy and anticipated future needs, while development is normally focused on a current need. To make things even more complicated, development project managers are normally evaluated only for meeting current project needs.

Well-functioning teams recognize this natural difference and work within both sets of goals to negotiate suitable solutions.

Architecture and planning, by design, have conflicting goals and standards to a one-time development project. Well-functioning teams recognize this natural difference and work within both sets of goals to negotiate suitable solutions.

However, this natural conflict can be difficult to manage when the stakes are high or the project is behind schedule. It is common in these situations for tempers to flare and professional differences to become personal ones. It's also common for both sides to retreat to their own groups, pitting developers against architects and vice versa. Once situations get this strained, it is very tempting to blow off steam by critiquing the other side, their work habits, their motivations and their competencies.

Stop Making it Personal

Team members should recognize the situation and take steps to defuse the conflict and the first thing they should do is *stop any personal criticism*. It is just fine to feel frustrated, but there is no good outcome for personal or professional attacks. Even if you feel you need some empathy or commiseration, any discussions are almost guaranteed to make their way back to the other party.

Of course there is a difference between "bad mouthing" a team member and escalating a debate to be decided by someone else. Any hint of personal tensions in the escalation is bound to work against you, so keep everything to the facts – costs, benefits, and risks around each proposal. In fact, you should know the cost, benefit and risk associated with all the proposals being considered. If you can articulate these, you'll be way ahead of those with a solution looking for a problem to be solved.

End Users Lose Respect When IT is Disrespectful

Another benefit to keeping your personal comments to yourself is that business users absolutely hate to sit through any type of architectural "religious" debate about some technical approach. They will tolerate a small amount of consideration of options, but nothing lowers confidence in the entire IT department more than an ongoing feud between the architects and the development staff. Keeping it all about the facts will keep issues from looking like turf wars.

I've also found a side effect of having highly-available models is that management and end users want to lean towards ensuring those models continue to give benefits to the organization. When I'm debating an approach with another team member whose components aren't as visible, management tends to side with the model-driven approach because they are most familiar with data models.

Finally...

Collaboration sometimes means giving up something you want in return for something the project needs -- and someone else wants. Keep the result of your project in focus without losing sight of long-term goals. If your team members have learned to respect your work due to your good collaboration skills, they will also see the value of your models.

About the Sponsor

IDERA

IDERA understands that IT doesn’t run on the network – it runs on the data and databases that power your business. That’s why we design our products with the database as the nucleus of your IT universe.

Our database lifecycle management solutions allow database and IT professionals to design, monitor and manage data systems with complete confidence, whether in the cloud or on-premises.

ER/Studio is the collaborative data modeling solution for data professionals to map and manage data and metadata for multiple platforms in a business-driven enterprise data architecture.

Whatever your need, IDERA has a solution.

About the Author

Karen López is a Senior Project Manager and Architect at InfoAdvisors. She has more than twenty years of experience in helping organizations implement large, multi-project programs.

InfoAdvisors is a Toronto-based project and data management consulting firm. We specialize in the practical application of project methods and tools. Our philosophy is based on assessing the cost, benefit, and risk of any technique to meet the specific needs of our client organizations.

InfoAdvisors offers data modeling training, including training for non-modelers such as business users, database administrators, project managers and developers.

We would like to help you add business value to your information management resources.

For more information, visit www.infoadvisors.com.

[1] Gerald Weinberg, *Secrets of Consulting: A Guide to Giving and Receiving Advice Successfully*, (New York: Dorset House Publishing, 1985), p. 56.