



White Paper

Data Migrations

Conducting your data's Digital Transformation

By John Peacock





Table of Contents

Introduction	3
Step 1: Analyze	3
Step 2: Plan	4
Step 3: Test	5
Step 4: Execute	6
Step 5: Verify	6
Conclusion	6



Introduction

A strong data foundation is critical to having an up-to-date, reliable, and efficient system architecture. Over time, a system is expanded upon in increments by multiple developers. The original data foundation can become less efficient, making data relationships overly complex. This lack of efficiency could also cause unused or redundant data or data format inconsistencies.

Whatever the issues, restructuring your data foundation is expensive and risky unless you're also able to change the pieces that rely on this foundation (i.e. applications, business intelligence tools, etc.). As a result, migration to a stronger data foundation is a critical first step of any digital transformation effort.

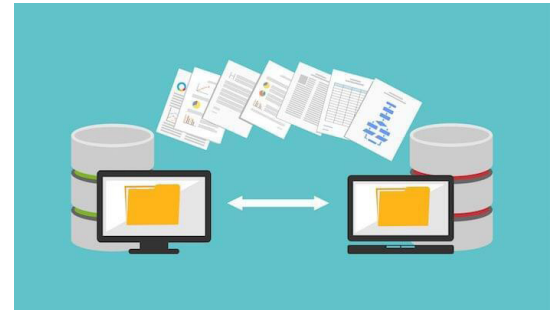


Image from Technology Advice ^[2]

Digital transformation gives us the opportunity to fix these issues with our data foundation. To do so, there are a few key points we need to address:

- How to transition to a newer foundation?
- What are the pitfalls to try to avoid?
- How does Appian affect the path forward?

There is no one “correct” solution to all these questions. Every business has their own needs and will have to analyze their situation to find their solution. However, here are some guidelines for beginning the journey into one of the most important aspects of digital transformation: data migration.

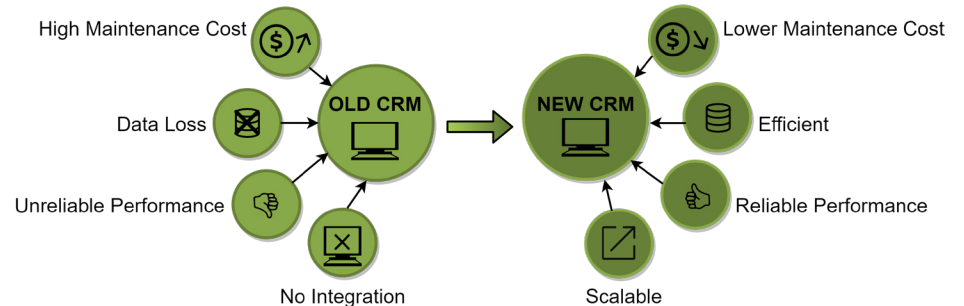
The process of doing data migration in the context of digital transformation can be daunting. The following breaks it down into 5 major steps to make it more manageable.

Step 1: Analyze

First, determine what flaws the current data foundation may have. Some common issues are redundant data, inconsistent formatting, and overly complex relationships. These data issues can sometimes be tricky to resolve. Consider an application with one table that had over 60 columns in the relational database which houses all the data for the application. The application started out much smaller, only needing a few fields, but as the application grew so did the data foundation backing it. No entry could be related to another, and there wasn't a separate table for common values, both of which caused their reporting system to be unwieldy and impossible to manage.



Using the single table implementation mentioned above as an exercise, the system could be an order tracking system where each row was a product ordered by a customer. Redesign the data foundation around the current issues to be the base of the data migration plan. The redesign can also introduce risks or issues if not executed carefully. In this example, we need to make sure we remove duplicate data, but we do not want to forget all the fields and relationships. For instance, let's say the table was broken up into 3 separate ones: customer, order, and products. Customer information should not be in the products table and vice versa, but the order table should have that relationship connecting them.



Going through and finding all the issues is crucial to making sure there is a solid data foundation. Analyzation can be a long process, but will prevent issues in the future and subsequently, help keep our digital transformation process more cost effective.

Step 2: Plan

Now that the analysis is done, start planning for what is needed out of the data foundation. The current issues and how to fix them are already known, so figuring out the needs of future applications is the next step. New applications tend to need additional fields for upcoming requirements or regulations, but it is also possible that there are data points that are not needed anymore. Removing these can help clean up the system but also makes the migration a little bit easier.

Reporting requirements are another important aspect to consider. The main questions to answer when considering reports are:

- What information is needed for our reports?
- Do we already have these data points, or are they new?
- If they are new, where do we place them?

A number of tools exist that facilitate reporting, including Appian. While Appian is a powerful digital transformation tool, and provides great value for end users, their role in data migration is usually limited. Like many such tools, Appian is great for creating visual reports, but in general the goal is to offload as much of the data manipulation to the database as possible. This can be accomplished through the use of views, materialized views, or stored procedures. Each one has their pros and cons and is going to be subject to the



system and application being built. Appian is the end goal when it comes to users interacting with data, not the tool for doing the migration. It is recommended to research the tools out there to find the best one that fits your business needs.

Step 3: Test

Testing is just as important within this process as any other software development life-cycle. Running mock migrations can help find any issues in our plan and fix them prior to using the actual production systems. Before testing can begin, there are several pieces that need to be set-up: a source environment, a target environment, production-like data, and test scripts for verification once everything is complete.

The source environment should be identical to the current production system. It should have the same tables and the same system architecture to be a proper test. Take the production-like data and insert it here. It is recommended to not use actual production data in most cases since it may have PII (personally identifying information), so either remove the sensitive elements of the data or use a tool to generate new data. Either way, the goal is to create a similar load size to the production environment.

The target environment is going to be based on what was created during the Analyze and Plan steps. Again, it should mimic the future production state in the same way the source mimics the current production state. Utilize monitoring on both systems to be able to record and analyze the load and stress being put on the system. This will help determine if any changes are necessary to the server architecture during this process.

Lastly, create test scripts for verifying all the data was correctly migrated. This can be done in a couple different ways:

1. Verification through Appian. If your application development is happening in parallel with this data migration effort, then this could be a good point to do testing with your application. Appian also has some easy-to-use tools for this type of testing, including using FitNesse For Appian.^[1] This is a testing framework built on top of Selenium and can be invaluable for doing smoke testing and regression testing.
2. Verification through database. If this route is chosen, create SQL scripts that will help verify that all the data was inserted, either through row counts or views.

Creating and running these tests may take a few iterations to get right, but once the validity of the migration has been verified, start preparing for running the actual migration in production.



Step 4: Execute

Once the test runs have been completed successfully, it is time to execute the actual migration with production data. However, before the migration is performed, backups need to be made of the target system in case there is a need for rollbacks. Thanks to all of the testing done previously, this should be a straight-forward step. However, make sure to avoid a crucial mistake: the target system should be isolated. It is possible that this data migration is happening in parallel with application development that has made it to production already. If so, that application should not push information to the database while the migration is in progress.

Step 5: Verify

The final step: verify that the migration was successful. The benefit of the earlier creation of the test scripts, either in FitNesse or SQL, will make this step much smoother. Running those scripts will confirm not only that the data was migrated to where it should be but that all the data was moved and, when necessary, reformatted. So, while this step may seem the “easiest”, it is just as crucial as any other. Should any issues be found, the backups made in the Execute step can be used to roll back to while the problems are fixed.

Conclusion

At this point, assuming everything went smoothly, the data migration process is complete. Finding the right solution requires investigation and a strong understanding of where you are and where you’re headed. Let’s recap the key take-aways:

- **Analyze your foundation.** What issues do you currently have: data redundancies? Too much overhead with maintenance? Figure out your pain points so you can directly address them with your new foundation.
- **Plan for your future state.** Don’t just address the issues found in your analysis, also look towards what your future application is going to require and mold your new foundation accordingly. This will mitigate risks as systems are continually worked on and updated.
- **Test your execution plan.** Make sure you iron out any kinks in the migration scripts.
- **Execute your migration plan.** When the production data foundation is ready and backups have been made, it is time to see the fruits of your careful planning.
- **Verify your data has been migrated correctly.** Theoretically any issues were found in the testing phase, but you can never be too careful!

Data migration is a complicated process, but the complexity and headache it could cause can be reduced by following the above steps. Planning and preparation at each phase is key for a successful migration.



Macedon is a recognized leader in intelligent automation and cloud data solutions. We have deep expertise with industry-leading technologies that we leverage to solve our clients' unique challenges.

Our hybrid roles achieve better solutions faster than traditional development teams.

Contact: (571) 526-4281
info@macedontechnologies.com

About the Author

John Peacock is an Appian Enterprise Architect and has been working at Macedon for ten years. He has experience managing and maintaining applications in Energy/Utilities, Financial Services, Government, Manufacturing, Services, and more with over 20 applications.

[1] Fitness for Appian

<http://community.appian.com/w/the-appian-playbook/97/automated-testing-with-fitness-for-appian>

[2] Scott, Tamara. "How to Perform a Successful Data Migration." Technology Advice, 23 May 2017

<http://technologyadvice.com/blog/information-technology/how-to-perform-successful-data-migration/>