Experience Report Test Data Management

Manage test data from different sources when replacing old applications

Version: 1.00

Author: Koch, Jörg

QIQ Qcentris Intelligent Quality GmbH
Berliner Straße 60
D-02826 Görlitz
Mail: infode@qcentris.com
Internet: www.qcentris.com

Geschäftsführer
Reto Züst
Peter Weber
Amtsgericht Dresden
HRB 91629
USt.-IdNr. DE 279276630

Volksbank Raiffeisenbank Niederschlesien eG
BLZ 85591000
Konto 4530152109
IBAN DE21855910004530152109
BIC GENODEF1GR1

Introduction

Normally, for existing IT systems which are already in productive use you can access after a while on a representative dataset. Such data sets providing an added value for the quality assurance of software development in future releases of these IT systems, because it requires often "only" an (possibly legally required) anonymization for using these data sets for software tests. But how to proceed with the generation of test data if existing applications are getting old and should be replaced by new applications which use other database architectures? An Experience Report.

Starting Situation

An older CRM system was to be replaced by a latest-generation system. Some specific, additional customer information were managed in two more peripheral systems. For each record a middleware linked the data of the different systems with each other. This was done by so-called *relations*.

The old CRM system included a data management for contact and address details (basic data). The two peripheral systems accessed these basic data, too. The new CRM system no longer contained a basic data management. In the future the basic data were managed by an additional peripheral system.

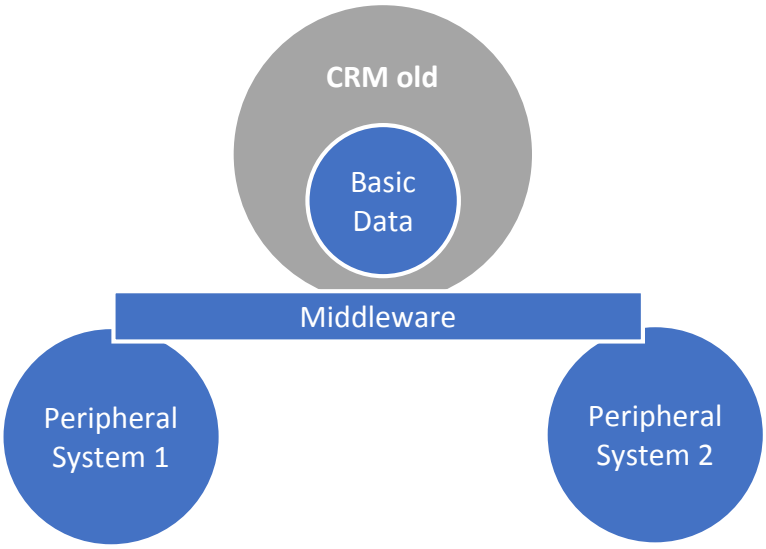

Figure 1: State of the old system landscape

The database architectures of the old and the new CRM system were different. So a data migration was required. In addition, the master data from the old CRM system must be extracted because in the future they were managed in the new additional, peripheral system.

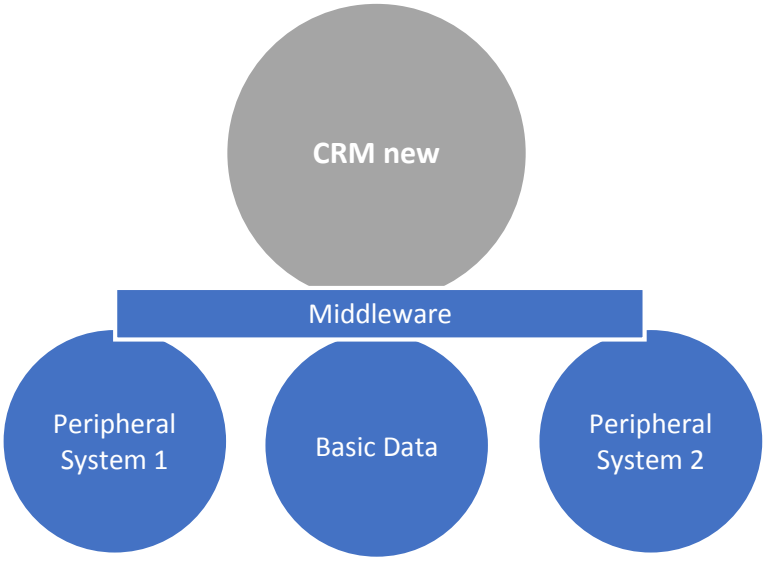

Figure 2: Planned restructuring of the system landscape

At the date of software development of the new systems, the new system landscape (Figure 2) and the new data architecture existed only on the development and test environments. In the production environment there was still the old structure (Figure 1). In consequence it was not possible to transport the production data to the test environment with a "simple" copy procedure.

Approaches to a solution

A short workshop resulted in two possible options for a solution:

Option 1:

Developing a test data generator which is able to generate respective records according to the desired expression.

Option 2:

Set up an additional system environment and convert/migrate the complete production data into the new database structure.

Decision

We chose for the following reasons the second option:

- Latest on Go-Live the complete production data must be migrated and merged. Now, we were able to test this procedure in advance.
- It was already a tool in use which was able to select specific data sets via appropriate search queries. After selection, it copied these data sets from the production environment on any other system environment. This had the advantage that not the entire existing data (about several terabytes) has to be copied, especially during the later software maintenance. Now, we developed already new queries for the new database structure during the software development which we could use for the software maintenance immediately after Go-Live. Another positive aspect of this tool was that specially defined anonymization could be made during the copying procedure.
- With an additional system environment as a data master in read-only mode we had automatically a backup of our test data. With it we were able to restore a defined state on each test environment after each test procedure.

Implementation

In the first step we merged the complete production data on the new created data master. In the second step we could distribute the corresponding records to the respective test environments after selection by queries (Figure 3).

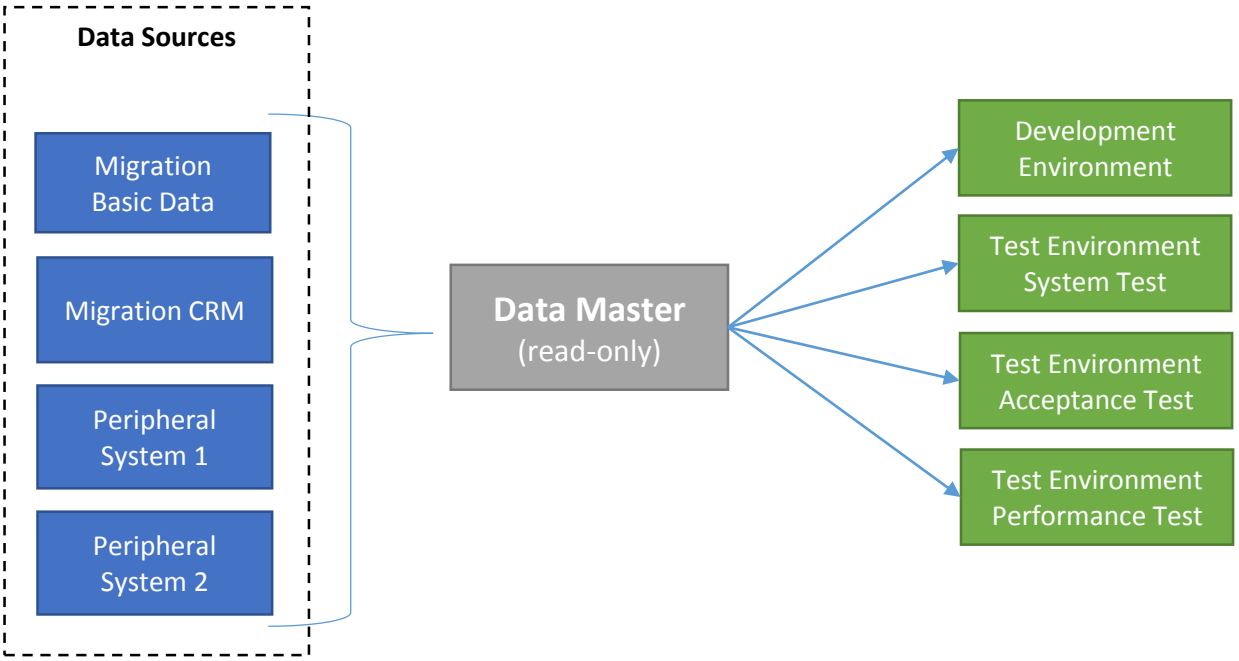

Figure 3: Merging and distributing the test data

Results

It was the right decision to choose the second option for implementation. We had identified several weak points in our doing which we had successfully optimized for the Go-Live:

- At the first creation of the data master there were many points where it finally had crashed. The causes were often just small technical things and quickly resolved. If these errors were occurred during the Go-Live, this would have caused unnecessary stress. But now we were well prepared and the Go-Live proceeded routinely.
- As part of the software development project, special daily, weekly and monthly jobs had to be adapted, which scanned the contract data in the CRM system and modified it if necessary. Now, on the dedicated performance test environment existed a representative data inventory, and we were able to make a sure statement about the processing times of the jobs.
- The planned tests of migration data could be combined with the system and acceptance tests. As a result the total expense were reduced.
- Now, we had a pretty good opinion what time is required for the several steps on Go-Live. As a result we had sure, practical values by planning the Go-Live.
- Certain technical activities we could execute in parallel during the Go-Live, without the performance of the entire system collapsed. We were able to evaluate this in advance during building the data master.

Initially it was assumed that the individual development and migration teams were able to align well among themselves in building the data master. But from the time where the application managers were included in setting up this data master, the need for coordination increased. Thus we met the important decision to deploy a dedicated Test Data Manager for coordination. Later in the project, he was responsible for the requirements of the individual system environment owners regarding their test data and coordinated the creation of the queries.

Conclusion

Test data management is not a luxury or "gimmick" but with professional and the project adapted tasks and measures it can significantly contribute for the success of the project. Moreover, it can provide a sustainable test data foundation for the future quality assurance during operation and the phase of the software maintenance!

……

The author Jörg Koch is Senior Consultant for Software Quality Assurance at QIQ Qcentris Intelligent Quality AG and has acquired a wealth of project experience, including in test data management.

QIQ QCENTRIS stands aside its customers as an independent, leading European consulting and services company with know-how in all aspects of the areas Quality, IT change management and testing aside. The company employs Test-, Quality- and IT-Change Experts in Switzerland, Austria, Germany and Egypt.