Data Migration from Legacy Systems to Modern Database

Sushma Velimeneti
Data Migration from Mainframe (Legacy) Systems to Modern Database

(SQL Server Database)

by

Sushma Velimeneti

A Starred Paper
Submitted to the Graduate faculty of
St. Cloud State University
in Partial Fulfillment of the Requirements
for the Degree
Masters of Engineering Management

August, 2016

Starred Paper Committee:
Hiral Shah, Chairperson
Ben Baliga
Balasubramanian Kasi
Abstract

Many businesses which use mainframe systems are unreliable these days and are facing the problem of data inconsistency and inaccuracy. Moreover the costs incurred in maintaining these systems keeps on increasing. As the competitors are migrating over to the new technologies and tools available in the market, it is necessary for these businesses to migrate to new environment which is efficient, consistent and reliable to stay in the market and the technologies used in the current environment have become obsolete or no longer serve the business purpose. This report summarizes the various methodologies, procedures and techniques involved in successfully migrating the data to modern systems. The practical work carried out for the capstone project is the development/building of application for data migration that is tailor-made for the client and their specific needs. In the process of data migration, we profile the existing data which helps in determining the structure and completeness of the data, in turn results in improved data quality. This report also contains data analysis that are carried on basis of profiled data and transforming the data accordingly to meet the new business needs. Agile software development life cycle is followed in the application building process as it uses iterative and incremental approach, bridging the gap between business owners and application developers as the end users are involved in each iterative phase. This report concludes with results that are obtained from successful data migration from the perspective of business needs and recommendations that could be implemented in the process of scaling of application
Acknowledgement

I would like to thank my advisor Dr. Hiral Shah for her support, guidance and cooperation for helping me in my capstone project. Who has always been a source of encouragement and knowledge for me, guided me in every step, and shared her knowledge. She has been most generous and understanding with her time to read this paper carefully and make insightful comments and suggestions.

I would also like to thank Dr. Ben Baliga and Dr. Balsy Kasi for their guidance and encouragement throughout the entire study.

I would also like to thank the Engineering Management Program for providing the resources.

I would like to thank my personal supervisor and all the staff of the Client for their support and guidance to carry out this project.

Last but not least, I would also like to thank my Parents and Husband and also my friends for all their guidance and support for the successful completion of my Master’s program.
# Table of Contents

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>List of Table</td>
<td>7</td>
</tr>
<tr>
<td>List of Figures</td>
<td>8</td>
</tr>
<tr>
<td>1. Introduction</td>
<td>9</td>
</tr>
<tr>
<td>Introduction</td>
<td>9</td>
</tr>
<tr>
<td>Problem Statement</td>
<td>9</td>
</tr>
<tr>
<td>Nature and Significance of the Problem</td>
<td>9</td>
</tr>
<tr>
<td>Objectives of the Project</td>
<td>10</td>
</tr>
<tr>
<td>Project Questions</td>
<td>10</td>
</tr>
<tr>
<td>Assumptions</td>
<td>10</td>
</tr>
<tr>
<td>Definition of Terms</td>
<td>11</td>
</tr>
<tr>
<td>Summary</td>
<td>12</td>
</tr>
<tr>
<td>2. Review of Literature</td>
<td>13</td>
</tr>
<tr>
<td>Introduction</td>
<td>13</td>
</tr>
<tr>
<td>Background Related to the Problem</td>
<td>13</td>
</tr>
<tr>
<td>Literature Related to the Problem</td>
<td>14</td>
</tr>
<tr>
<td>Data Migration Implementation</td>
<td>19</td>
</tr>
<tr>
<td>Legacy Systems–Mainframe Database</td>
<td>20</td>
</tr>
<tr>
<td>Modern Database Application–Microsoft SQL Server</td>
<td>22</td>
</tr>
<tr>
<td>Statistics and Data Migration Success Rates</td>
<td>23</td>
</tr>
</tbody>
</table>
Chapter | Page
--- | ---
Literature Related to the Methodology | 23
User Acceptance Testing (UAT) | 26
Quality Assurance Test | 30
Summary | 32
3. Methodology | 33
   Introduction | 33
   Design of Study | 33
   Data Analysis | 37
   Data Collection | 38
   Budget | 39
   Project Timeline | 40
   Summary | 40
4. Data Presentation and Analysis | 41
   Introduction | 41
   Data Presentation | 41
   Data Analysis | 42
   Project Timeline | 42
   Summary | 43
5. Results, Conclusion, and Recommendations | 44
   Introduction | 44
   Results | 44
Project Questions Answered ................................................................. 46
Conclusion .......................................................................................... 48
Recommendations ............................................................................. 49
References .......................................................................................... 50
List of Table

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Project Timeline</td>
<td>40</td>
</tr>
</tbody>
</table>
### List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Data migration frame work</td>
<td>20</td>
</tr>
<tr>
<td>2. Overview of Agile methodology</td>
<td>26</td>
</tr>
<tr>
<td>3. UAT and its components</td>
<td>27</td>
</tr>
<tr>
<td>4. Process of software quality assurance</td>
<td>31</td>
</tr>
<tr>
<td>5. Iterative methodology for data migration</td>
<td>37</td>
</tr>
<tr>
<td>6. Overview of stages involved in data collection</td>
<td>39</td>
</tr>
<tr>
<td>7. Data profiling of customer table in legacy system</td>
<td>41</td>
</tr>
<tr>
<td>8. Project timeline</td>
<td>43</td>
</tr>
<tr>
<td>9. Performance enhancement</td>
<td>44</td>
</tr>
<tr>
<td>10. Down time of the application is reduced</td>
<td>45</td>
</tr>
<tr>
<td>11. Compression in database size in SQL server when compared to mainframe</td>
<td>45</td>
</tr>
<tr>
<td>12. Performance enhancement</td>
<td>48</td>
</tr>
</tbody>
</table>
Chapter 1: Introduction

Introduction

The client is one of the largest healthcare carrier located in the United States. They have an excellent IT infrastructure to support their health services. The aim of the project was to define the next generation Healthcare Payer administration platform that would enable a cohesive end-to-end Health Plan administration platform, including; Account Implementation, Member Enrollment, Product Definition, Provider Networks, Contracting and Reimbursement, and Claims Adjudication.

This project was carried out at corporate location of client located at Minnetonka, MN. In order to build the next generation Healthcare Payer administration platform client needed to migrate the data from Mainframe System to Modern Database (SQL Server) which would help the organization in achieving more efficient, flexible, effective and realistic environment to work in.

Problem Statement

Due to the heavy maintenance, operational and performance costs involved in maintaining the Mainframe databases and their inflexibility to support new business initiatives, the client wanted to migrate the data from these various mainframes to a common consolidated repository under SQL Server 2012 database. The existing client applications then will start communicating from SQL Server 2012 instead of Mainframes

Nature and Significance of the Problem

The Mainframe once held an unquestionable lead in reliability, availability, serviceability and security. As new modern technologies evolved, they were more efficient,
innovative and flexible than the Mainframes. Due to high maintenance costs involved, using Mainframes was not cost effective. This resulted in a need to shift to Microsoft Application Platforms along with its additional features in-turn helps in the business growth.

**Objectives of the Project**

Objectives/project goals were:

1. Data profiling and analysis of the existing data
2. Migrate the data from legacy system to modern database system
3. Lessen the operational and maintenance cost.
4. Provide advantage/superiority over the competitors
5. Enhanced customer support
6. Providing a highly scalable and performance efficient solutions.

**Project Questions**

The following are the project questions that were answered after the project was completed:

1. What is the need for organizations migrate their databases?
2. What are end benefits of data migration for the business owners?
3. What can be done to make sure that only high quality data is transferred during data migration?
4. What was the performance improvement of application after the data migration?
5. What was the percentage of impact on the Down Time of the application?

**Assumptions**

The main assumptions of this project were:
1. Automated data migration is carried out from legacy to modern database application using prepackaged development tools.

2. Data Migration from legacy system to modern applications is carried out by considering the business requirements.

3. The Extraction and Conversion process will be tested prior to the production.

4. Testing of the applications are done within the given timeframe.

5. All the issues arised during each phase of the project is reported timely to Business for remediation.

6. All the users will perform their tasks within the given migration slots.

**Definition of Terms**

- **ETL**—stands for extraction, transformation, and loading. The process of extracting data from source systems and bringing it into the data warehouse is commonly called ETL.

- **SQL**—stands for Structured Query Language. SQL is used to communicate with a database. According to ANSI (American National Standards Institute), it is the standard language for relational database management systems.

- **RDBMS**—RDBMS stands for Relational Database Management System. RDBMS is the basis for SQL, and for all modern database systems such as MS SQL Server, IBM DB2, Oracle, MySQL, and Microsoft Access.

- **Legacy Systems**—A legacy system, in the context of computing, refers to outdated computer systems, programming languages or application software that are used instead of available upgraded versions.
• SQA—Software quality assurance (SQA) is a process that ensures that developed software meets and complies with defined or standardized quality specifications.

• SDLC—stands for Software Development Life Cycle. A Software Development Life Cycle is essentially a series of steps, or phases, that provide a model for the development and lifecycle management of an application or piece of software.

• UAT—stands for User Acceptance Testing. We know what testing is, acceptance means approval or agreement. User in the context of a software product is either the consumer of the software or the person who requested it to be built for him/her (client).

Summary

In this chapter the need for the client to upgrade to newer database applications and to migrate the data for achieving more efficient flexible, effective and realistic environment to support their health services, nature and significance of the problem, objective, and goals of the project and assumptions of the project implemented are discussed. Background related to the problem, literature related to the problem and methodology used for implementing the project will be discussed in the next chapter.
Chapter 2: Review of Literature

Introduction

The project is mainly based on the planning of migration methodology and implementing smooth migration for the whole project for providing best service in health services. The importance of the risk free strategy is very critical as the client has zero tolerance to stall the business activities for the data conversion and data discrepancies too. In today’s modern world due to emerging technology new database applications available in more customized and organized manner. Because of the rapidly improving technology for computers and changes in business requirement the normal life cycle for a database application is between 5 and 10 years. After this time the application’s ability to store, process and retrieve data will not match the needs of the organizations or the technology on which it operates will have become obsolete. Therefore the organizations will be required to implement a new database application which leads to data migration from legacy systems to modern applications.

Background Related to the Problem

Data migration is the process of transferring data from one system to another. Data migration differs from data movement, data integration and conventional ETL (extract, transform and load) respectively by migrating to a new environment, by being a one-time process and by maintaining the consistency of usage. The success rate of data migration projects is low, although not as low as in other software projects The time it takes to perform a data migration if one only considers the extract and load phases is a function of the size of the data objects, the speed in the transferring medium and the degree of parallelism. The data migration project concerned in this report is between different databases and the data
migrations concerns an invoice system for a company whose business expertise is to provide services for domain model, this section starts with the theoretical background behind databases and domain names.

**Database.** A database is a collection of information that is organized so that it can easily be accessed, managed, and updated. In one view, databases can be classified according to types of content: bibliographic, full-text, numeric, and images. The collection is managed by a Database Management System, DBMS.

The DBMS provides the users with:
- The ability to create databases and decide how the data will be structured in the database.
- The ability to query a database for the wanted data, and to modify that data.
- The ability to store very large amounts of data over a very long period of time. i.e., Persistence storage.
- The ability to control the access to data from many users at the same time.

**Domain migration.** Moving data from one system, or domain, to another without compromising security or losing any of the data. Domain migration happens when servers are upgraded and the data (including any authentication and authorization information) must be moved to a new system.

**Literature Related to the Problem**

**Database migration checklist.** Hudicka (1998) describes about the importance of migrating legacy systems to modern applications. In his article he explained about the strategy and complications of data and its nature to be considered while migrating to modern systems.
Russom (2006) explains about the necessity of migration, different solutions based on type of migration, different technical tools to select based on the type of source and target systems. In his article he states that ETL (extract, transform and load) is the preferred technology for data migration.

Jones (2014) explains about the data migration process and risks to be considered during migration planning. When embarking on a data migration project we need to identify the risks and activities well in advance. Due to the risk involved in data migration may delay the deployment of new database applications leads to increased cost, data loss, and downtime. So it is effective to identify the risks involved in migration during planning phase itself. This describes how to identify the potential risks involved at various phases.

**Project initiation.**

- Create a high level project plan.
- Create a standard project document such as a project controls, risk registers, job descriptions, change management report, project progress reports, Acceptance criteria, project controls.
- Define the infrastructure requirements such as hardware and software tools like data profiling tool, reporting software, project management software, etc.
- Get approvals from legacy source systems to analyze the source systems without impacting the business and resolve any security issues related to accessing data.

**Scale the project.** (Determine size of migration project: time, cost, labor skills, and risk to the organization).

- Estimate work days and actual costs for each phase and total migration.
• Ensure a budgeting tool is included in the migration planning process.

• Maximize the efficiency of the resource allocation.
  
  o Senior technical staff—difficult tasks.

**Pre-migration planning.**

• Assess the viability of migration with pre-migration impact assessment.

• Involve both business and IT teams before the migration kicks off.

• Consider the data security and governance issues.

• Identify the key resources and time allocation.

• Determine the project delivery and estimate by considering risks.

• Create the appropriate training documentation and design a training plan.

• Define project approach and work flow.

**Scope the migration project.** (To avoid load-rate bottleneck problems and to ensure successful migration.)

• Deciding on strategic delivery vehicles for applications: even if using desktop virtualization, consider presentation layer impact.

• Use pre-migration application compatibility assessment to determine readiness.

• Create a detailed scoping report for the migration by considering business requirements.

• Scoping document should clearly include what function should be carried out in a new application, will existing application should maintain or decommission, data
quality management practices like data cleansing to be taken or not, the information in new application into or out from any other applications.

**Establish an environment for design, implement and test.** (Determine an environment (Database) Design and Test).

- Focus efforts with simple “traffic light reporting” in red/amber/green.
- Gain deep insight into database application behavior by analyzing business requirements and establish the environments to carryout migration.
- Create a detailed data dictionary, database design, conceptual, logical data model, etc. for initiating the data migration from legacy systems.
- Choose the best options for application deployment based on the status of the application compatibility testing.
- Manage regular application deployment and ongoing change Compared to the time of manual testing, this step alone can reduce migration testing time very much.

**Create migration plan.**

- Define phases of the project.
- Define tasks that must be completed in each phase.
- Define pilot group.
- Define key milestones and set realistic timelines.

**Develop optimized migration process.**

- Map out processes and remove duplicated effort and pain points.
- Optimize the process using a workflow methodology (ideally specialized).
- Know where applications are in the process, who owns it, when it will be ready.
Implement targeted testing. Test applications with targeted testing in minutes per application instead of manual testing taking days. Focus expert testers on actual problem areas:

- Targeted testing uses pre-migration application compatibility to guide testers on what is broken.
- Eliminate “smoke testing”—light testing that does not fully interrogate the application.
- Cut testing time and budget; eliminate bottlenecks in application testing.

Refresh and regression testing. At some point in the migration project, the source data will need to be refreshed. Typically data is refreshed prior to each testing cycle and certainly prior to go-live. Two challenges to be considered during a refresh:

- Create a process to apply all data manipulations including manual updates/additions.
- Design a set of reconciliation reports for regression testing after a refresh.

Go-live strategy. Whether as part of a phased implementation or a “big bang” go-live strategy, it is important to plan for business operations during, and immediately following the period of go-live.

- Ensure that you have a good estimate of the time required to load and validate the new systems.
- Define how data will be synchronized between systems during any period of parallel operations.
• Get an agreement from the project steering group on what criterion define the end of the pilot phase.

**Data Migration Implementation**

Bryan (2009), for implementation of data migration, needed to setup a development environment through which we can have connections to the database containing the data to be extracted and a means of creating the logical models of the source and target databases and a means of manipulating data and connections to the database into which the data to be loaded. Gamji (2012) explains about the staging process from source system to target system. In that article, he explained the detailed process of extraction, transformation and loading process involved in migration of a database.

**Data extraction and data cleansing.** Data extraction is the process of retrieving the data from the source database into a logical model of the source application. Figure 1 describes about various steps involved in data migration from legacy systems and briefly shows the various steps involved in data migration. This is usually done by writing SQL queries on the source database tables or using a migration tool that allows you to specify what data needs to be extracted.

Data cleansing can be carried out directly in the source database, in the migration scripts or by manipulating the data. Data cleansing comprises of domain checking, enforcing integrity constraints and de-duplication

**Data transformation.** Data transformation is the process of transforming the data in the logical models of the source applications into logical model of the target applications. The
main operations used in the transformation process are mapping, filtering, separating and combining.

**Data loading.** Data loading is the process of moving the information in the logical model of the target application into the physical database tables. This is usually achieved by writing SQL insert or update scripts.

![Conversion and Migration Framework](image)

*Figure 1. Data migration frame work (ELT Solutions, 2016).*

**Legacy Systems–Mainframe Database**

Wu et al. (1997) explains about the approach to mission-critical legacy systems migration and gives information related to legacy systems and why legacy systems need migration to modern systems.

From time to time various Business organizations implement new Database Application System to replace the functionality currently delivered by one or more legacy systems. Complications arise when there is an attempt to take the information currently
maintained by the legacy system and transform it to fit into the new system. More often, the data structure of the legacy systems is different from the new application being implemented, and that difference is not just limited to the table names, field names or attributes or sizes. The types of databases are different and diverse, or the entity relationship definition in the new system are not compatible with the older legacy application. To the business organizations all the data being held in the legacy system remains critical for their business and decision making.

To bring the legacy system data to the new application some Data migration must take place, where an initiative, separate or concurrent with the implementation of the new application, is undertaken to convert data from one structural form, used by the legacy application to the structural from required by the newer application.

Often in a Data Conversion process, one would tend to think that any two similar systems that maintain the same sort of data, as they are doing very similar functions should map from one to another without much trouble. But that is not really the case as:

- In Legacy systems, historically, data integrity checks were not strictly enforced, leaving orphan data.
- Theoretical design differences exist between hierarchical and relational systems.
- Legacy data may require some data cleansing.

**Mainframe database-legacy system.** Transvive, Inc. (2011) describes the migration project lifecycle including assessing the current database, planning, designing, implementing and managing the migrated environment. This article explains about the assessment of legacy systems before migration.
Most legacy systems hosted on mainframes are custom-built. A couple of decades ago, they were developed using third-generation programming languages, and monolithic architecture and procedural styles. Business logic and data access components were bundled together and rolled out in functional release cycles. This approach created significant redundancy in technical and functional services—including data and security access checks. Inflexibility was a problem. Point-to-point communication applications had to adapt to fixed data exchange interfaces. The technical skills required to transform legacy systems were not commonly available. Enterprises of all sizes are adapting by Re-engineering all, or portions, of their home-grown mainframe legacy systems into open source or other available distributed technology stacks and mid-range platforms. And also migrating and absorbing business processes into packaged solutions available in the market.

**Modern Database Application—Microsoft SQL Server**

**Microsoft SQL Server** is a relational database management system developed by Microsoft. As a database, it is just a software product whose primary function is to store and retrieve data as requested by other software applications, be it those on the same computer or those running on another computer across a network (including the Internet). There are at least a dozen different editions of Microsoft SQL Server aimed at different audiences and for different workloads (ranging from small applications that store and retrieve data on the same computer, to millions of users and computers that access huge amounts of data from the Internet at the same time). The primary query languages are T-SQL, ANSI SQL.
Microsoft makes SQL Server available in multiple editions, with different feature sets and targeting different users like enterprise edition, Datacenter, Standard edition, Web edition, Business intelligence etc. Microsoft SQL Server has high security and compliance. It is a consistent data platform on premises of the cloud. It helps in accessing the data with familiar tools like Excel, Notepad, etc. (This information is retrieved from www.wikipedia.com and www.techopedia.com.)

**Statistics and Data Migration Success Rates**

According to the answers to the Bloor research (Howard & Potter, 2007) and other studies concluded that as many as 64% of the data migration projects fail, either by going over time or going over budget or, as in many cases, both time and budget. The main reason for the overruns in time and budgets appear to be the difficulties to successfully plan and scope the project. The data migration success rates may seem like very poor results and may be an indication on that a more standardized methodology for data migration is needed.

While a majority of the data migration projects ran over both time and budget, the main reason they ran over budget was because of the time overruns. And it's easy to understand that a time overrun in most cases, if not all, results in an overrun in the budget. Migration has to be carefully planned and executed to minimize production downtime and maintain data integrity and database performance, because many companies have locations worldwide and have to be operational 24 x 7.

**Literature Related to the Methodology**

This section briefly describes about the literature related to methodology used for data migration from legacy systems to modern systems.
IBM Global Technology Services (2007) outlines the strategy and implementation of data migration. It briefly outlines the impact of data migration on business process, Migration Plan, Design of migration, implementation and Maintenance of data migration. (Oracle successful data migration, 2011) documented the various process to follow for migrating the legacy systems to modern applications.

**Agile methodology.** An introduction to Agile methods describes the methodology in Agile software development lifecycle. Agile development provides opportunities to assess the direction throughout the development lifecycle. Figure 2 explains the lifecycle of agile development. This is achieved through regular cadences of work, known as Sprints or iterations, at the end of which teams must present a final product. By focusing on the repetition of abbreviated work cycles as well as the functional product they yield, agile methodology is described as “iterative” and “incremental.”

**Step-wise breakdown of agile methodology.** Singh (2008) provides the detail overview of Agile methodology. The Agile methodology provides frequent and early opportunities to see the work being delivered, and to make decisions and changes throughout the development project. The user of this methodology gains a strong sense of ownership by working extensively and directly with the project team throughout the project. Development is often more user-focused, likely a result of more and frequent direction from the customer

**Planning:**

- Identification of stakeholders and sponsors
- Infrastructure Requirements
- Security related information and gathering
- Service Level Agreements and its conditions

**Analysis:**
- Capturing of Stories in Parking lot
- Prioritize stories in Parking lot
- Scrubbing of stories for estimation
- Define Iteration SPAN (Time)
- Resource planning for both Development and QA teams

**Design:**
- Break down of tasks
- Test Scenario preparation for each task
- Regression Automation

**Implementation:**
- Coding
- Unit Testing
- Execution of Manual test scenarios
- Defect Report generation
- Conversion of Manual to Automation regression test cases
- Mid Iteration review
- End of Iteration review

**Evaluation and closure:**
- Pilot Launch
- Training
- Production Launch
- SLA Guarantee assurance
- Review SOA strategy

*Figure 2.* Overview of Agile methodology (Inadev, Ingenuity Unleashed, 2016).

**User Acceptance Testing (UAT)**

The goal of User Acceptance Testing is to assess if the system can support day-to-day business and user scenarios and ensure the system is sufficient and correct for business usage. It is also called beta testing, application testing, and/or end user testing—is a phase of software development in which the software is tested in the “real world” by the intended audience or a business representative. Figure 3 describes various components involved in UAT. While the technical testing of IT systems is a highly professional and exhaustive process, testing of business functionality is an entirely different proposition (retrieved from https://www.develop.com/useracceptancetests).
Need of user acceptance testing. Developers and functional testers are technical people who validate the software against the functional specifications. They develop an understanding of the requirements and develop/test the software (here is the importance of domain knowledge). This software is complete according to the functional specifications but there are some business requirements and processes which are known to end users only are either missed to communicate or misinterpreted. UAT plays an important role in validating if all the business requirements are fulfilled or not before releasing software for the market use. Use of live data and real use cases make UAT testing an important part of the release cycle (retrieved from https://www.techopedia.com/2/29662/software/software-testing/why-do-we-need-user-acceptance-testing-uat).
Many businesses who have encountered massive losses due to post-release issues have realized the importance of successful UAT. The cost of fixing defects after release is many times greater than fixing it before.

**UAT team composition.** Mainly end users of the software should be involved in UAT. The team can be comprised of beta testers or customers should select UAT members internally from every group of organization so that each and every user role can be tested.

**Major challenges of UAT and mitigation plan to overcome these.** It does not matter if you are part of a billion dollar release or a startup team, it is mandatory to overcome these UAT challenges for delivering successful software for the end user.

**UAT environment setup and deployment process:** Carrying out UAT on same environment used by functional test team will certainly end up overlooking real world use cases. Also crucial testing activities like performance testing can’t be carried out on test environment with incomplete test data. Separate production like environment should be setup for UAT.

Once the UAT environment is separated from test environment the release cycle has to be effectively controlled. Uncontrolled release cycle may leads to different software versions on test and UAT environment. Valuable UAT time is wasted by not testing software on latest version. Not to mention the time required for issue tracking on incorrect software version.

**UAT test planning:** UAT should be planned with clear acceptance test plan in the requirement analysis and design phases. In strategy planning, set of real world use cases should be identified for execution. It is very important to define test objectives for UAT
testing as complete test execution for large application is not possible in UAT phase. Testing should be carried out by prioritizing critical business objectives over other aspects.

UAT is carried out at the end of the testing cycle. Obviously it is the most critical period for the software release. Delay in any of the previous stages of development and testing eat up UAT time. Improper test planning, in worst cases, leads to overlap between system testing and UAT. Due to less time for UAT and pressure to meet deadlines, software is deployed to UAT environment even though functional testing is not completed. UAT goals can’t be achieved in such situations.

UAT test plan should be prepared and communicated to team well before beginning UAT. This will help them for test planning, writing test cases and test scripts and creating UAT environment.

*Handling new business requirements as incidents/defects:* Ambiguities in requirements get caught in UAT phase. UAT testers find issues arising due to ambiguous requirements (by looking at the complete UI which wasn’t available during requirement gathering phase) and log it as a defect. Customer expects these to be fixed in current release without considering the time for change requests. If timely decision is not taken by project management on these last-minute changes this could lead to the release failure.

*Unskilled testers or testers without business knowledge:* If there is no permanent UAT test team, company select UAT staff from various internal departments. Even if this staff is well familiar with business needs, if they are not trained for the new requirements being developed they cannot perform effective UAT. Also non-technical business team might face many technical difficulties executing UAT test cases. Also assigning testers at the end of
the UAT cycle do not add any value to the project. Little time to train UAT staff can significantly increase the chances of UAT success.

**Improper communication channel:** Communication between remote development, testing and UAT team is more difficult. Email communication is often very difficult when you have an offshore tech team. A small ambiguity in incident reports can delay its fix for a day. Proper planning and effective communication are critical to effective team collaboration. Project teams should use web based tool to log defects and questions. This will help to distribute work load evenly and avoid reporting duplicate issues.

**Asking functional test team to perform UAT:** There is situation worse than asking functional test team to perform UAT. Customers offload their responsibility to test team due to lack of resources. The whole purpose of UAT gets compromised in such cases. Once the software goes live, end users will quickly spot the issues which are not considered as real world scenarios by functional testers. Solution to this is to assign UAT to dedicated and skilled testers having business knowledge.

**Quality Assurance Test**

Software quality assurance (SQA) is a process that ensures that developed software meets and complies with defined or standardized quality specifications. SQA is an ongoing process within the software development life cycle (SDLC) that routinely checks the developed software to ensure it meets desired quality measures (retrieved from http://www.softwaretestinghelp.com/successful-user-acceptance-testing/).

SQA helps ensure the development of high-quality software. SQA practices are implemented in most types of software development, regardless of the underlying software
development model being used. In a broader sense, SQA incorporates and implements software testing methodologies to test software. Rather than checking for quality after completion, SQA processes tests for quality in each phase of development until the software is complete. With SQA, the software development process moves into the next phase only once the current/previous phase complies with the required quality standards. SQA generally works on one or more industry standards that help in building software quality guidelines and implementation strategies. These standards include the ISO 9000 and capability maturity model integration (CMMI).

![Diagram of Software Quality Assurance](image)

*Figure 4. Process of software quality assurance (Software Testing & QTP, 2016).*

**Software safety.**

- Defined as a software quality assurance activity that focuses on identifying potential hazards that may cause a software system to fail.
• Early identification of software hazards allows developers to specify design features to can eliminate or at least control the impact of potential hazards.

• Software reliability involves determining the likelihood that a failure will occur, while software safety examines the ways in which failures may result in conditions that can lead to a mishap.

Summary

In this chapter, literature related to the problem, study and methodology used were discussed. Framework of the study, data collection process, techniques used to analyze the data will be discussed in the next chapters.
Chapter 3: Methodology

Introduction

This chapter provides a detailed description of the methodology used for the study. Data migration methodology is an in-depth process where we bring in different stakeholders of application at various phases of the data.

Design of Study

Agile software development life cycle methodology was followed in process of application development. In this methodology, application is built in iterations which is reviewed by the end business owner at the end of each iteration to bridge the gap between end business user and developers. The migration was planned to be carried in two steps which contain different phases which can move back forth. Figure 5 depicts the iterative methodology of data migration process followed.

Scoping. In this phase identify the potential data sources from which information needs to be migrated to the new application. In this phase a scoping document is prepared, which defines the questions and decisions taken and finally arriving at a data migration strategy that needed to be followed. Some of the major questions factors that are considered while designing the scoping document are”

a. Is the information currently held in the legacy system, to be used in our new application?

b. Does the historic data held for more than 10 years needed?

c. Does the legacy systems contain large amount of duplicate data which needs to be normalized?
d. Is the information used for showing any kind of reports?

After the scoping document has been approved the next phase Data migration planning.

**Planning.** This phase was carried out in multiple stages where feedback from our previous stages has been incorporated into our feature following which the project timeline got changed.

This stage will have following phases:

1. Requirement Gathering: In this stage, all the requirements for building new target database has been gathered, by identifying the different data sources of the existing mainframe database. Also gathered are all the business rules and domain knowledge to analyze the source data.

2. Landscape Analysis: After the requirements were gathered a landscape analysis has been done, to gain create an overview of source and target systems in which working of each system has been understood, and how the data is structured/modeled and the interaction between different models.

3. Data Assurance: In this phase a review/validation of the existing data in our source systems has been done, to determine whether it is suitable for transforming it to target systems with the help of landscape analysis. This step consists of two sub steps.
   - **Data Profiling and Mapping:** This document was created to help the people to carry out the process of data migration. This document maps the fields in the source and target systems prepared by profiling whether the historical data
present in source systems is suitable for the migration to target systems and also deleting the historical data which is of no use.

- **Data Cleansing**: In this step, checking the quality of the data in the source systems like whether the data is available, complete and in correct format has been done. If not modified the data to suit to the target systems and also deleted the data not needed for migration.

4. Core Migration:

- **Migration Design**: This step defined the transformation logic to move data from source to target systems according to the mapping definitions present in the mapping document.

- **Testing Design**: This step verified whether the migration design is aligning with the business rules by performing various kinds of testing like unit testing, Integration testing and functional testing etc.

- **Migration Development**: Once migration design is tested and approved, the process of the Migration development took place where the data was extracted, transformed and loaded into target systems.

Extract, Transform and Load: ETL is a process where we extract, transform and load the data that is gathered and mapped by performing Data profiling and mapping into our new target database.

- Extract-In this step the data was extracted from the existing upstream data sources according to the mapping document created from the above stages and made available for the further processing.
➢ Transform- In this step the extracted data was transformed, based on business rules to be applied on them like filtering the data by selecting only the columns which are required, joining the data from different tables and performing aggregate operations and calculation of new fields based on the extracted data.

➢ Load- Finally in this step the transformed data was loaded into our target database system: i.e., SQL Server database.

   All these data migration steps were performed using various ETL Tools like Informatics, Data stage, SAP data integrator.

   • Testing Development (User Acceptance Testing): All the data that is migrated was rigorously tested by the same test cases that were used before to test old database like

       1. Number of records that are fetched by a particular script?
       2. Is the extracted data properly formatted in the target system?
       3. Are the source data fields mapped accurately to the target database fields?
       4. Is the integrity and quality of data is maintained?

   Once the application passed the QA, it was sent for business owners to test.

   • Business Sign Off: After the application successfully went through QA. The test results were sent to the business owner to provide signoff.

   • Go Live and Production Support: After the successful completion of the core migration testing phase and various sprints, the application was released for production and further enhancements were made for the application based on changes in the business requirements.
The agile development life cycle has been followed and all corrupt migrations were caught at early stages of development which were be rectified in the parallel sprints by changing the project timeline. Agile development life cycle helped to identify misinterpretations in between business owner and development team and were rectified in parallel sprints.

*Figure 5. Iterative methodology for data migration (ELT Solutions, 2016).*

**Data Analysis**

Before the data migration begins analyzed the source data that is strictly needed to be migrated.
Steps in Data Analysis:

1. Analyze how the data is structured and organized in our legacy systems by understanding who uses it and what they use for. Make use of the existing documentation of legacy systems for the sake of gathering information.

2. Analyze how our target database systems is prototyped and suggest if any modifications are required depending on the analysis of legacy systems.

3. Create a detailed mapping document containing mapping of the fields in legacy systems to that of the fields in target systems by performing various techniques of data profiling using profiling tools.

4. While profiling the data, determine whether there is a requirement of data cleansing. Data cleansing means de-duplicating the data, conversion of data types and removing any data anomalies if found.

Data Collection

After the data has been analyzed the process of data collection from our existing legacy systems would begin.

Steps in data collection.

1. Find all the required data sources needed to be migrated to the new system.

2. Once the required data sources are identified, design a staging database which is mirror copy of the source databases.

3. While creating the staging database, create the required objects with the same name and data type as in our legacy systems.

4. Ensure that appropriate storage space is allocated for all the required objects.
5. Ensure the data access to all the roles confers with legacy systems access roles.

6. Map the data collected to target systems using the data mapping document created in the process of data analysis.

7. Verify the data samples using the reports in our legacy systems.

Figure 6 shows the basic overview of stages involved in the process of data collection.

![Figure 6. Overview of stages involved in data collection.](image)

**Budget**

The entire project was completed within the cost provided by the client. No extra costs were incurred during the development of this project. All the future activities are also expected to be completed in budget.
Project Timeline

The following milestones are planned for the project:

Table 1

<table>
<thead>
<tr>
<th>ESTIMATED DATES OF THE MILESTONES</th>
<th>STAGES</th>
</tr>
</thead>
<tbody>
<tr>
<td>August 2015</td>
<td>Requirement Gathering</td>
</tr>
<tr>
<td>August 2015 – September 2015</td>
<td>Data Mapping document</td>
</tr>
<tr>
<td>September 2015 - November 2015</td>
<td>Migration Design, Development and</td>
</tr>
<tr>
<td></td>
<td>User acceptance testing.</td>
</tr>
<tr>
<td>December 2015 - February 2016</td>
<td>Go live and Production support</td>
</tr>
<tr>
<td>March 2016</td>
<td>Defense</td>
</tr>
</tbody>
</table>

Summary

This chapter provided a look at the methodology used, carried an in-depth analysis of the roles and implementation of different steps used in for making a successful data migration plan. In the next chapter various data analysis techniques that are employed while collecting the data were discussed in detail.
Chapter 4: Data Presentation and Analysis

Introduction

This chapter will focus on the various models and techniques used to collect the data and also outlines various analysis techniques used.

Data Presentation

Business rules were applied on the data that is profiled from legacy systems for transforming it to target systems.

Figure 7. Data profiling of customer table in legacy system.

Following are some of the observations collected from the above figure which shows data profiling of customer table in legacy system:

1. Customer Id can be used as primary key in target system
2. Address field (ADDR2) contains null values which in reality doesnot happened as every customer needs to have the address. So, removed the data as it is incoisistent while migrating to target system
3. According to business rules address should be stored interms of CAPITALS in target system, so while tranforming the data to target system, ensured that the
address is stored in CAPITALS. Figure 7 explains about the data profiling of the legacy system.

**Data Analysis**

Analysed the data mainly using various profiling and other techniques. Data profiling helped in both understanding the data and discovering the anomalies in data.

Some of the techniques used were

1. **Column profiling:** Identified data types, range of values and format of the data stored in each column and used this information to provide insights into business.

2. **Row Profiling:** Using this technique, anlaysed the data across the rows to indentify whether the integrity constraints like (Primary key, Unique Key, Foreign key, Default, etc.)

3. **Redundancy Profiling:** Identification of any information in the system is duplicated at other places of the source data systems has been done using this technique

4. **Data Consistency:** Identified whether data is consisent across the multiple tables where we store similar information.

All of these techniques were used in determining relevance, accuracy, completeness of the data in source systems.

**Project Timeline**

The project was started in August 2015 and is almost reached completion stage.

Figure 8 below is the project timeline showing various phases of development lifecycle of the project and also describing the milestones achieved in the progress of the project.
Figure 8. Project timeline.

Summary

This chapter covered various analysis and data profiling techniques used for migrating the data according to business requirement. Next chapter outlines the results and conclusions.
Chapter 5: Results, Conclusion, and Recommendations

Introduction

This chapter will focus on the project results eventually obtained, the conclusion from these results and lastly the recommendations that are suggested during the implementation of the project.

Results

The iterative development lifecycle was employed for data migration which helped in achieving project objectives on time and in budget. Involving the business owners and end users by following the data migration methodology was also one of the major factor in achieving the goals. The results of the migration effort can mainly be presented by the three main progress indicators:

1. **Enhanced Performance**: Due to the data migration, new application performance has been boost up by 65% on an average. Figure 9 shows how the average time taken for actions has been decreased before and after data migration.

![Reviewer 1](image)

*Figure 9. Performance enhancement.*
2. **Lessen Down Time:** Data which is migrated weekly in older system makes the system to be down on weekends. Real time entry in new system reduced the downtime due to its efficient storage and data compression. Figure 11 shows how much the database size has been reduced in SQL server when compared to Mainframe.

![Figure 10. Down time of the application is reduced.](image)

![Figure 11. Compression in database size in SQL server when compared to mainframe.](image)
3. **Real Time Report Analysis:** Due to real time data storage all the reports are up with live data. Due to the modernized database various kinds of reports are added for analyzing the data.

**Project Questions Answered**

The following are the project questions that are answered:

1. **What is the need for organizations migrate their databases?**

Following are some of the parameters that show the need for organizations to migrate their databases

- System performance has been boost up by 65 percent on an average.
- Reduced down time due to real-time data entry.
- Real time analysis reports resulting in serving customers in a better way.
- Scalability and adaptability to new challenges in business swiftly.
- Reduced database size by storing and handling of complex data efficiently.
- Improved security and reliability of the application.
- Improved interoperability as the newer technologies can be easily integrated with the modern systems.

2. **What are end benefits of data migration for the business owners?**

Following are some of benefits that the end business owners obtained

- Improved decision making due to real time report analysis.
- Improved services and supports giving edge over competitors.
- Decreased operational and maintenance costs involved due to heavy costs involved in maintaining multiple data sources in legacy systems.
• Leveraging new functionality and features of the modern systems.

3. What can be done to make user that only high quality data is transferred during data migration?

The older the systems, the deeper are issues in data quality. When the existing data in source systems was transformed to target systems, all the inaccuracies, unknowns, redundant and duplicate content in fields became apparent in data profiling stage like analyzing Address field in customer table described in Figure 7. Removed these data anomalies through various profiling techniques like: row, column and redundant profiling and also ensured the data integrity by enforcing foreign key, primary key and unique key constraints in Data cleansing stage.

4. What was the percentage of performance improvement of application after the data migration?

Due to the data migration, new application performance has been boost up by 65% on an average. This is mainly because of the effective indexes created on large tables and effective data storage mechanism employed in target systems removing redundant data. Figure 12 shows how the average time taken for actions has been decreased before and after data migration.
Figure 12. Performance enhancement.

5. **What was the percentage of impact on the Down Time of the application?**

Data which is migrated weekly in older system makes the system to be down on weekends. Real time entry in new system reduced the downtime due to its efficient storage and data compression techniques. Down time is also reduced because of upgrade of a software component such as the operating system etc. Down time of the application is reduced by almost 85%.

**Conclusion**

The migration of the legacy systems to modern systems (SQL Server) helped in enhancing the performance of the application, reducing down time and also in providing real time report analysis. The project started as per the plan and considered any business requirement changes in the process and completed on time and in budget. Effective data migration strategy helped achieving the projects milestones without many hurdles on time and
also helped in achieving project goals/objectives. On the whole, project goals and targets were successfully met.

**Recommendations**

Although the project was successfully completed, 24/7 support for at least 3 months is recommended so that if any crash occurs due to data migration so that end user never gets affected. There is a scope for adding lot of new reports to analyze the data by using the features of newly upgraded system.
References


