Implementation of Agile Methodology in a Data Conversion Project Using the Informatica ETL Tool

Shashank Suroju

Follow this and additional works at: https://repository.stcloudstate.edu/mme_etds

Recommended Citation
Suroju, Shashank, "Implementation of Agile Methodology in a Data Conversion Project Using the Informatica ETL Tool" (2016). Culminating Projects in Mechanical and Manufacturing Engineering. 57.
https://repository.stcloudstate.edu/mme_etds/57

This Starred Paper is brought to you for free and open access by the Department of Mechanical and Manufacturing Engineering at the Repository at St. Cloud State. It has been accepted for inclusion in Culminating Projects in Mechanical and Manufacturing Engineering by an authorized administrator of the Repository at St. Cloud State. For more information, please contact rwexelbaum@stcloudstate.edu.
Implementation of Agile Methodology in a Data Conversion Project

Using the Informatica ETL Tool

by

Shashank Suroju

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

August, 2016

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

This project was implemented in an American national telecommunications company located in the Texas, United States of America which provides High speed internet (HSI), Direct to home (DTH) service, technical support to all its customers located across 28 states in the USA. This project proposal will focus on implementation of agile methodology in the conversion of data from source database to target database. This project involves in analyzing the source table information, applying the transformation logic on how to populate the data and loading the data into the target table. After the data is loaded to the target database we can verify the target count and see if the data is loaded correctly or not.
Acknowledgments

Firstly, I would like to express my deepest gratitude to the professors of my committee Dr. Hiral Shah, Dr. Ben Baliga, and Prof. Balsy Kasi who have given me an opportunity and enough time to complete this project. The knowledge which I have gained from the professors at St. Cloud State University really helped me in applying the principles and complete the project successfully on time. Furthermore, a special thanks goes to my manager Swain Corey and my team members who helped me in gathering the required and correct information which was very helpful. Finally, I would like to thank my family and friends who have given me a great support in fulfilling my dream to complete the study at St. Cloud State University.
# Table of Contents

List of Tables .................................................................................................................. 6
List of Figures .................................................................................................................... 7

Chapter

1. Introduction .................................................................................................................... 9
   Introduction .................................................................................................................... 9
   Problem Statement ...................................................................................................... 10
   Nature and Significance of the Problem .................................................................... 10
   Objective of the Project .............................................................................................. 11
   Project Questions ........................................................................................................ 11
   Limitations of the Project ........................................................................................... 11
   Definition of Terms ..................................................................................................... 12
   Summary ....................................................................................................................... 12

2. Background and Review of Literature ................................................................. 14
   Introduction .................................................................................................................. 14
   Background Related to Problem .............................................................................. 14
   Literature Related to the Problem ............................................................................ 14
   Literature Related to the Methodology .................................................................... 17
   Summary ....................................................................................................................... 20

3. Methodology ............................................................................................................... 21
   Introduction .................................................................................................................. 21
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Design of Study</td>
<td>21</td>
</tr>
<tr>
<td>Data Analysis</td>
<td>47</td>
</tr>
<tr>
<td>Budget</td>
<td>48</td>
</tr>
<tr>
<td>Timeline</td>
<td>48</td>
</tr>
<tr>
<td>4. Data Presentation and Analysis</td>
<td>50</td>
</tr>
<tr>
<td>Introduction</td>
<td>50</td>
</tr>
<tr>
<td>Data Presentation</td>
<td>50</td>
</tr>
<tr>
<td>Data Analysis</td>
<td>51</td>
</tr>
<tr>
<td>5. Results, Conclusion, and Recommendations</td>
<td>56</td>
</tr>
<tr>
<td>Introduction</td>
<td>56</td>
</tr>
<tr>
<td>Results</td>
<td>56</td>
</tr>
<tr>
<td>Project Questions</td>
<td>56</td>
</tr>
<tr>
<td>Conclusion</td>
<td>57</td>
</tr>
<tr>
<td>Recommendations</td>
<td>57</td>
</tr>
<tr>
<td>References</td>
<td>58</td>
</tr>
</tbody>
</table>
# List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Agile in LHS Process</td>
<td>25</td>
</tr>
<tr>
<td>2. Target Table VZ3CZ5.IFIDDI Structure</td>
<td>27</td>
</tr>
<tr>
<td>3. VZ3CA5.CBST128 Table Structure</td>
<td>28</td>
</tr>
<tr>
<td>4. VZ3CA5.CBST129 Table Structure</td>
<td>29</td>
</tr>
<tr>
<td>5. VZ3CA5.CBST0014 Table Structure</td>
<td>31</td>
</tr>
<tr>
<td>6. Logic to Populate the Columns Other Than Address Fields</td>
<td>41</td>
</tr>
<tr>
<td>7. Timeline of the project</td>
<td>49</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>1.</td>
<td>General overview of waterfall model</td>
</tr>
<tr>
<td>2.</td>
<td>Overview of LHS process</td>
</tr>
<tr>
<td>3.</td>
<td>Complete mapping design of IFIDDI</td>
</tr>
<tr>
<td>4.</td>
<td>Source qualifier transformation</td>
</tr>
<tr>
<td>5.</td>
<td>Router transformation</td>
</tr>
<tr>
<td>6.</td>
<td>Group one address columns logic in expression transformation</td>
</tr>
<tr>
<td>7.</td>
<td>Complete workflow of group one records</td>
</tr>
<tr>
<td>8.</td>
<td>Group two address columns logic in expression transformation</td>
</tr>
<tr>
<td>9.</td>
<td>Complete flow of group two records</td>
</tr>
<tr>
<td>10.</td>
<td>Logic to populate the non-address columns in expression transformation</td>
</tr>
<tr>
<td>11.</td>
<td>Complete mapping design of sprint 2</td>
</tr>
<tr>
<td>12.</td>
<td>Smart logic for maining_address</td>
</tr>
<tr>
<td>13.</td>
<td>Smart logic for service_address</td>
</tr>
<tr>
<td>14.</td>
<td>Address fields populated from IFISM table</td>
</tr>
<tr>
<td>15.</td>
<td>Work flow monitor</td>
</tr>
<tr>
<td>16.</td>
<td>Target data address fields in VZ3CZIF5.IFIDDI table</td>
</tr>
<tr>
<td>17.</td>
<td>Total number of records fetched from source</td>
</tr>
<tr>
<td>18.</td>
<td>Total number of records loaded into the target table</td>
</tr>
<tr>
<td>19.</td>
<td>Total number of records which have address fields populated</td>
</tr>
</tbody>
</table>
20. Total number of records for which address fields are not populated

.............................................................. 55
Chapter 1: Introduction

Introduction

This project was implemented in an American national telecommunications company located in the Texas, United States of America. Focusing mainly in rural areas, this company was ranked as fifth largest provider of digital subscriber line and sixth largest local exchange carrier in USA. Other than telephone service, this company also provides High speed internet (HSI), Direct to home (DTH) service, technical support to all its customers located across 28 states in the USA. The company has some 3 million residential and business voice subscribers. It also has 1.8 million broadband Internet customers and more than half a million satellite video subscribers. The company's top three markets in terms of subscribers are West Virginia, Indiana, and New York.

In February, 2015, Company had announced a consensus with another telecom giant under which this company will purchase the other company`s wireline, broadband and FiOS operations in California, Texas and Florida. This acquisition had doubled company`s size after it was complete. The project was finished in April 2016.

Agile software development technique, which refers to the group of methodologies, where requirements and solutions evolve as the project progresses, was used in the conversion and migration of data from source database (selling company) to the target database (buying company). Informatica ETL tool was used in conversion and migration of data.
Problem Statement

The company has previously used the waterfall approach in the conversion project which has its own disadvantages such as too much of rework is required, high risk and uncertainty. This traditional approach is not suitable for the conversions projects where the requirements gets changed frequently.

Nature and Significance of the Problem

This organization has followed the water fall approach for the previous conversions. Water fall model is very simple to use and in this model the next phase will begin only after the current phase gets completed. Each and every phase will not get overlapped. A rigorous and thorough upfront analysis of requirements is done before starting the development stage. Once the requirements gets fixed and are clearly understood then the development activities begin. And after this the testing and maintenance activities takes place. In this type of model, the customer interaction is very less and the cost of rework is very high in case of any failures, because everything has to be built from the beginning. This type of model is very good for smaller projects and also for the projects where requirements are specific and certain and also understood well.

But, for a company where the requirements gets changed frequently, waterfall approach is not the best practice to be used. So the company has used agile methodology in migration of the data from source (selling company) database to target (buying company) database. And the conversion of data was done by using the Informatica tool. Due to short iterations in agile methodology, changes can be
implemented easily. Since the sprint end dates were fixed and did not change, there was no delay in delivery and was delivered on time.

**Objective of the Project**

The main objective of the project was to implement the agile methodology in a data conversion project and move the data from source database to target database and

- Decrease the overall time of implementation by at least 40%.
- Cut cost of the overall implementation.
- Make the target database available with correct and accurate data.

**Project Questions**

The project after successful completion has answered the below questions:

1) What were the improvements that can be made by using this approach?
2) What was the overall impact at the organization level by following this approach?
3) What were the target deliverables that were achieved by the implementing the agile approach?

**Limitations of the Project**

The two main limitation which were encountered while executing the project were time and no reusable artifacts. Since the main objective of the project was to cut down the implementation time, the time which was allocated was less. And also since the agile methodology was implemented for the first time, there were no previous
artifacts/models which can be used as reference. However, the data from the source is accurate and was loaded into the target accurately.

**Definition of Terms**

*IF File*: It is known as the Interface file (the target table) where the target data is loaded after extracting from the source.

*DSLAM port*: Digital Subscriber Line Access Multiplexer located in telephone exchange used to connect DSL and HIS.

*SME*: Subject Matter expert.

*SQL*: Structured Query Language.

*DPI*: It is the server box where the data is collected from target tables.

*Scrum master*: Person who is a facilitator of the project and manages the process.

*Stakeholder*: People who want and need a product because of which the development team builds one.

*RPG Programming*: It is the IBM high-level programming language.

*Library/Schema*: It is a repository in the database where all the objects like tables, views and synonyms are located.

**Summary**

This chapter has briefed many important aspects of the project like giving an insight to what the actual problem was and how it affects in real time, main objective of the project, list of questions that were answered after the successful completion of project, limitations of the project and, finally, the definition of all the terms that were
used in this project. Chapter 2 will cover the background and review of Literature associated with the current project.
Chapter 2: Background and Review of Literature

Introduction

This main focus of this chapter was provide the background related to problem, literature related to the problem and review the literature related to the methodology which was used to solve the problem.

Background Related to Problem

This problem was identified in a company which has its business operations located in major cities of USA. Around 4000 business and development professionals have worked on this project, with the development location located at Dallas, Texas.

The Waterfall Model is one of the old & earliest method of system development. Until recent times water fall model is the widely used methodology but it became less popular because of being too rigid to meet the customer`s requirements (Center for Technology in Government, 2003). So in order to meet the customer`s requirements there was a constant search for a new process in developing system applications. And eventually waterfall model is attributed in providing the theoretical basis for new methodologies (Center for Technology in Government, 2003).

Literature Related to the Problem

In the present generation software has an important role in any field like business, education, telecommunications, etc. Success mainly depends upon which software they are using, and because of the rapid development in technology, companies need to be up-to-date in the software they are using and that software
should meet all the companies’ requirements. The ever increasing competition between the companies have created a challenge for the right and correct software development methodology (Czarnacka-Chrobot, 2010).

In order to face this challenge, many software development models have been introduced. The very first model that was introduced in software development process is the waterfall model. It was introduced in 1970 and is a sequential development model in which all the development of a software goes through different phases in sequence. The developed software will go into the next phase only when the current phase is completed and documented and this will become the input for the next phase. A pictorial representation of this is shown in Figure 1.

In general, waterfall model has six different phases which are:

a) Requirement analysis: It is the first step and includes a detailed understanding of client requirements, the end software product functions which it will perform, and any external systems that it should be compatible with. There are many ways to gather requirements like customer interaction, use cases, etc.

b) Design: This stage mainly defines the hardware and software architecture, choosing the programming languages, resource management and interface connectivity (Melonfire, 2006). Considering all these factors one or more design specification document is developed which is used as an input for the next stage.
c) Implementation: This is the step where actual software product is constructed. It includes development team who use different technologies to construct the software product. The output is a software product build according the pre-defined standards which is debugged and tested.

d) Testing: An individual testing team is dedicated to test the software product. They test the product both at component level and whole as a system integrated level by writing test cases to test the product. There are different types of testing which include alpha testing, beta testing, system testing, system integration testing and user acceptance testing. The output is the final software product which has minimum number of errors.

e) Installation: Once the software product is tested and is ready for use by the customer, it will be installed at customer site or released into the market for customer use.

f) Maintenance: This stage includes correcting any errors discovered while in production or making any modifications to the developed software product. These changes are requested by the customer and are known as change requests.

Two main challenges for any organization are they need their business software on time and within the budget. And also, at the time of development, the organizational requirements may change as there will be changes in the business process, which will become very problematic for the development teams to implement the changes in the middle of the development activity and it is very expensive to implement these
changes as well. And thus developing a software becomes late and over budget (Awad, 2005).

So a new process has been introduced known as AGILE development methodology which can overcome these problems.

**General Overview of "Waterfall Model"**

![Waterfall Model Diagram]

*Figure 1. General overview of waterfall model.*

**Literature Related to the Methodology**

Due to the recent advancement in the field of internet there are huge changes in the business requirements and the traditional models like waterfall were not able to accommodate the changes owing to high budget constraints and hence failed. Because of these disadvantages of the waterfall process a new process has been
introduced which is the agile methodology and is an alternative to traditional process, which is typically used in software industry. Agile software development have been introduced because they are very flexible in incorporating the required changes and deliver the quality software product quickly (Livermore, 2007).

It is a group of software development methodologies where requirements and solutions evolve as the project progresses. Different types of methodologies are Extreme Programming, Feature Driven Development, and Scrum (Collier & Highsmith, 2003; Rising & Janoff, 2000). For this project we used the scrum methodology which is an incremental iterative process. Agile methodology breaks the whole project into small parts whose period varies from a single week to 6 weeks. These short periods are known as iterations or sprints which have minimal planning and are not objected for a long–term planning (Rising & Jarnoff, 2000). Each iteration is completed by the team (can be a single person), and goes through a full SDLC (Software Development Life Cycle), including requirement gathering, design, development, unit testing, system testing. The final output of the iteration might not have full functionality of the project but the goal of each iteration is to have a release at the end.

The main functionalities are built in the initial iterations. This concept is referred to as delivering which is “potentially shippable” (Rising & Janoff, 2000). This indicates that the product can be delivered to the customer with main functionalities even though all the functionalities of the product are not built in a particular release.
The additional functionalities can be built on top of it without changing what has already been coded.

According to (Abrahamsson, Salo, Ronkainen, & Warsta, 2002) a method is said to be agile if it has the following software development properties.

1) Incremental: small software release with rapid cycles.
2) Cooperative: customer and developers are in close communication.
3) Straightforward: easy to learn and well documented.
4) Adaptive: that can accommodate changes at any stage in the development.

There are different processes in agile methodology and scrum is the most popular methodology of introducing agility due to its simplicity and flexibility. In recent years, the software industry has witnessed many changes and the requirements change continuously to meet the challenges at the time of software development (Rising & Janoff, 2000). It provides a great support to the present software development activities. Scrum methodology will have different management activities and will help in identifying the development deficiencies.

**Informatica PowerCenter (9.5).** Informatica PowerCenter (9.5) is the most powerful Enterprise Data Integration products tool which is developed by Informatica Corporation. Informatica PowerCenter jumpstarts and enhances the data projects and will deliver the data to the clients up to five times faster than the other integration tools. Informatica PowerCenter is an ETL tool which is used in extracting the data from source database, transforming the data and loading the data into the target database. The main challenges with the traditional approach of loading the
data involves writing complex queries, connecting and pulling the data from different source systems. Informatica ETL tool will provide a ready-made solution for these kind of problems. ETL is a three step process which involves:

a) The EXTRACTION which involves in analyzing the sources tales and gathering the data from the sources tables.

b) The TRANSFORMATION which involves the cleansing of data.

c) The LOAD involves the loading of the data into dimension and fact tables.

The data which comes from source database is raw data and might contain duplicates, errors and will be inconsistent which can lead to making poor decisions (Informatica Corporation, 2013). Informatica power center is an ETL tool which is developed by Informatica Corp and the components of it are used to extract the raw data from different sources, transform the raw data and load the data into target database to build enterprise data warehouse (Informatica Corporation, 2013).

Summary

This chapter has covered the background related to problem, literature related to the problem, literature related to the methodology and also explained the theory of agile methodology.
Chapter 3: Methodology

Introduction

This chapter mainly focuses on the design of study, data collection and tools and methods used to analyze the data of the project.

Design of Study

This section describes in detail about the framework of the project.

Conversion process overview. Conversion process was divided into two sides:


b) Right Hand Side (RHS): RPG coding and DPI (Company’s target application) loading. (OUT OF SCOPE).

The Left Hand Side process has utilized the seller application data as source data, extracted the data, transformed the data and loaded the data into IF files shown in Figure 2. The Right Hand Side has used this IF files as source data and populated the corresponding DPI files.
The incoming data from the selling company was divided into four divisions:

a) **CUSTOMER**: Which has data related to the customers such as accounts, billing, directory labels, treatments, etc.

b) **PLANT**: Which has the data related to inside plant and outside plant like cables, ports, DSLAM ports, etc.

c) **PRODUCTS**: Which has information related to different products the company sells such as promotional offers, etc.

d) **SERVICE ORDERS**: Which has the information related services like setting up a new connection, renewing a connection, etc.

*Figure 2. Overview of LHS process.*
My area of work was in the Left Hand Side process in directory labels under customer division. **Extract** the data from the Oracle (Database) and AS400 (database), apply the business rules, **Transform** the data and **Load** the data in the IF files (interface files). Informatica tool was used to perform this ETL process.

**What is a DIRECTORY LABEL?** It was one of the IF file which captures the directory delivery address and directory quantities necessary to the customer. This will allow the directory book to be delivered to the appropriate customer location.

The development team had worked with the Frontier SMEs from the Residence and Business areas to match source data set up to DPI format. At necessary intervals, these mapping specs and their outputs had been communicated to the appropriate business teams and other conversion team representatives for feedback and for final sign off.

**Agile in LHS process.**

1) Project manager/Architect was identified. He was responsible for the design/model of the tables.

2) Project scrum master was identified and had decided to use the agile approach which was supported by stakeholders and team members

3) We had decided to finish the project in four sprints and the sprint size as 4-5 weeks which is shown in Table 1. But the size of the sprint can be changed as the project progress and deliverables met.
4) The main requirement of the project was to collect the data from the source database and load the data into the target database. The data was extracted from the below mentioned source tables.
   a. CBST0128
   b. CBST0129
   c. CBST0014

5) Below mentioned were the different resources identified to complete the project;
   a. Stakeholders/Business
   b. Scrum master
   c. Team member

6) Based on the inputs given by the stakeholders/business users, we had decided the information which needs to be pulled from the source tables.
Table 1

**Agile in LHS Process**

<table>
<thead>
<tr>
<th>Project Planning</th>
<th>Completed tasks</th>
<th>Delivered product</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tasks Planned</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Develop an architecture for the project such as creating the target tables, indexes on the tables, adding key constraints to the tables etc.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sprint 1: 5 weeks</th>
<th>Completed tasks</th>
<th>Delivered product</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tasks Planned</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Build the target table with address fields being populated</td>
<td></td>
<td></td>
</tr>
<tr>
<td>a) developed the SQL query to join the source tables 128,129</td>
<td></td>
<td></td>
</tr>
<tr>
<td>b) developed the SQL query to join cbst0014 tables of all the 3 regions TX,FL,CA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>c) Created the CATXFLCOMB table by joining the 128,129 tables for all the three regions TX,FL,CA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>d) Joined the tables CATXFLCOMB and CBST0014 comb to build CBSSFIDDI which is used as a source to load the target</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Loaded the data into the target table VZ3IFS.IFIDDI with address columns being populated</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sprint 2: 4 weeks</th>
<th>Completed tasks</th>
<th>Delivered product</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tasks Planned</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Build the target table with remaining columns populated</td>
<td></td>
<td></td>
</tr>
<tr>
<td>a) developed the SQL query to join the source tables 128,129</td>
<td></td>
<td></td>
</tr>
<tr>
<td>b) developed the SQL query to join cbst0014 tables of all the 3 regions TX,FL,CA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>c) Created the CATXFLCOMB table by joining the 128,129 tables for all the three regions TX,FL,CA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>d) Joined the tables CATXFLCOMB and CBST0014 comb to build CBSSFIDDI which is used as a source to load the target</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Loaded the data into the target table VZ3IFS.IFIDDI with remaining columns being populated.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sprint 3: 5 weeks</th>
<th>Completed tasks</th>
<th>Delivered product</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tasks Planned</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Implement the smart logic to make sure address 1 field and address field 2 gets populated</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Developed the logic to break the billing address/service address field into two such that the address field is broken down correctly.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>The target table is populated with Address1 and address 2 fields being populated correctly.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sprint 4: 4 weeks</th>
<th>Completed tasks</th>
<th>Delivered product</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tasks Planned</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Build the target table with address data being pulled from VZ3IFS.IFISM table</td>
<td></td>
<td></td>
</tr>
<tr>
<td>a) Changed the logic in the SQL query to join the CBSSFIDDI table with VZ3IFS.IFISM to pull the address fields from IFISM table.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>b) Pulled the data for the remaining columns using the previous logic.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>The target table address fields are populated using IFISM address fields and the remaining fields are populated using IFISB columns.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Project closure</th>
<th>Completed tasks</th>
<th>Delivered product</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tasks Planned</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Production warranty bug fixes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Production warranty bug fixes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Production warranty bug fixes</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Data collection and analysis:**

1) **Planning phase:** The data architect has created the below mentioned target table:
VZ3IF5.IFIDI (Target table): Below mentioned was the target table structure which has a total of 21 columns and all the details like length and description of each column are explained in Table 2.

The target table was created in the library: VZ3IF5 with the name IFIDI. The query which was used to build the target table is:

```
CREATE TABLE VZ3CZIF5.IFIDI (IDDICID CHAR(10) NOT NULL DEFAULT ",
IDDIPST CHAR(1) NOT NULL DEFAULT ",IDDIEXEC CHAR(6) NOT NULL DEFAULT ",
IDDIKDT CHAR(8) NOT NULL DEFAULT ", IDDIRES NUMERIC(1, 0) NOT NULL DEFAULT 0,
IDDIDLQ NUMERIC(5, 0) NOT NULL DEFAULT 0, IDDIMLQ NUMERIC(5, 0) NOT NULL DEFAULT 0,
IDDIOMA CHAR(1) NOT NULL DEFAULT ", IDDIKSN CHAR(15) NOT NULL DEFAULT ",
IDDIFRN CHAR(15) NOT NULL DEFAULT ", IDDIKF CHAR(1) NOT NULL DEFAULT ",
IDDIAD1 CHAR(30) NOT NULL DEFAULT ", IDDIAD2 CHAR(30) NOT NULL DEFAULT ",
IDDIYTY CHAR(20) NOT NULL DEFAULT ", IDDISAB CHAR(20) NOT NULL DEFAULT ",
IDDIZCD NUMERIC(9, 0) NOT NULL DEFAULT 0, IDDIORG CHAR(20) NOT NULL DEFAULT ",
IDDIACC CHAR(25) NOT NULL DEFAULT ", IDDIKT CHAR(53) NOT NULL DEFAULT ",
IDDIUDDT CHAR(8) NOT NULL DEFAULT ")
RCDFMT
IDDIKMT;
```
### Table 2

**Target Table VZ3CZ5.IFIDDI Structure**

<table>
<thead>
<tr>
<th>Sno</th>
<th>Field Name</th>
<th>Field</th>
<th>Length</th>
<th>Description</th>
<th>Description</th>
<th>Default/Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>IDDICID</td>
<td>CONVERSION ID</td>
<td>Default to VZ3CZ</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>IDDIPST</td>
<td>POSTED FLAG</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>IDDIEXC</td>
<td>ISBNPA</td>
<td></td>
<td>ISBNXX</td>
<td>A (6)</td>
<td>SECONDARY NPA/NNX</td>
</tr>
<tr>
<td>4</td>
<td>IDDILN#</td>
<td>ISBLIN</td>
<td>A (4)</td>
<td>SECONDARY line #</td>
<td>LINE</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>IDDICDT</td>
<td>ISBCDT</td>
<td>A (8)</td>
<td>Connect Date</td>
<td>CONNECT DATE</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>IDDRES</td>
<td>ISBRES</td>
<td>A (1)</td>
<td>Secondary Sequence #</td>
<td>RESEQUENCE</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>IDDIDLQ</td>
<td>ISBDLQ</td>
<td>A(2)</td>
<td># of Compact Directories</td>
<td>DIRECTORY LABEL QUANTITY</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>IDDIMLQ</td>
<td>--</td>
<td>--</td>
<td>Midi Label QUANTITY</td>
<td>No longer a product. Default to ‘0’</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>IDDIOMA</td>
<td>--</td>
<td>--</td>
<td>Exception Address</td>
<td>OTHER MAIL ADDRESS</td>
<td>Leave it blank</td>
</tr>
<tr>
<td>10</td>
<td>IDDILSN</td>
<td>ISBLNM</td>
<td>Other Last Name</td>
<td>OTHER LAST NAME</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>IDDIFRN</td>
<td>ISBFNM</td>
<td>Other First Name</td>
<td>OTHER FIRST NAME</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>IDDIONF</td>
<td>LASTNM</td>
<td></td>
<td>FIRSTNM</td>
<td>Other Name Format</td>
<td>OTHER NAME FORMAT</td>
</tr>
<tr>
<td>13</td>
<td>IDDIAD1</td>
<td>ADDR1</td>
<td>A(30)</td>
<td>Other Address 1</td>
<td>OTHER ADDRESS 1</td>
<td>***address to be broken into 2 fields. Location 1-30 2nd field 31-60.</td>
</tr>
<tr>
<td>14</td>
<td>IDDIAD2</td>
<td>ADDR2</td>
<td>A(30)</td>
<td>Other Address 2</td>
<td>OTHER ADDRESS 2</td>
<td>***address to be broken into 2 fields. Location 1-30 2nd field 31-60.</td>
</tr>
<tr>
<td>15</td>
<td>IDDICTY</td>
<td>CITY</td>
<td>A(20)</td>
<td>Other City</td>
<td>OTHER CITY</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>IDDISAB</td>
<td>STATE</td>
<td>A(20)</td>
<td>Other State</td>
<td>OTHER STATE</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>IDDIZCD</td>
<td>ZIPCD</td>
<td>O(9)</td>
<td>Other Zip Code</td>
<td>OTHER ZIP CODE</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>IDDIORG</td>
<td>ISBORG</td>
<td>Origin Data</td>
<td>ORIGINAL DATA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>IDDIACC</td>
<td>ISBACC</td>
<td>Original Account ID</td>
<td>ORIGINAL ACCOUNT ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>IDDIKT</td>
<td>ISBCKT</td>
<td>Circuit ID</td>
<td>Circuit ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>IDDICUST</td>
<td>ISBMI4</td>
<td>Customer Key</td>
<td>Customer Key</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
2) **Sprint 1: 6 weeks.** After talking to the business and the client SME’s, the source tables from which we can pull the data were identified. The source tables were:

A) **STEP1: Identified the source tables.**

1) **VZ3CA5.CBST128:** The table structure is described in Table 3. This table had the information related to the customer account number and **DIR_GRP_SEQ_NO.** This information was used to classify whether we have to pull the BILLING address or SERVICE ADDRESS.

Table 3

**VZ3CA5.CBST128 Table Structure**

<table>
<thead>
<tr>
<th>Column_Name</th>
<th>Data_Type</th>
<th>Column_Length</th>
<th>Default_Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CUST_ACCT_NO</td>
<td>CHARACTER</td>
<td>10</td>
<td>No default</td>
</tr>
<tr>
<td>DIR_GRP_SEQ_NO</td>
<td>DECIMAL</td>
<td>5,0</td>
<td>No default</td>
</tr>
<tr>
<td>CAP_CONTROL_NO</td>
<td>DECIMAL</td>
<td>7,0</td>
<td>No default</td>
</tr>
<tr>
<td>CAP_IND_LVL_VAL</td>
<td>CHARACTER</td>
<td>1</td>
<td>No default</td>
</tr>
<tr>
<td>COMSET_SEQ_NO</td>
<td>DECIMAL</td>
<td>5,0</td>
<td>No default</td>
</tr>
<tr>
<td>LST_ACCT_TYP_CD</td>
<td>CHARACTER</td>
<td>1</td>
<td>No default</td>
</tr>
<tr>
<td>LST_PUB_STAT_CD</td>
<td>CHARACTER</td>
<td>3</td>
<td>No default</td>
</tr>
<tr>
<td>LETTER_SEQ_OPT</td>
<td>CHARACTER</td>
<td>1</td>
<td>No default</td>
</tr>
<tr>
<td>OMIT_ADDR_OPT</td>
<td>CHARACTER</td>
<td>1</td>
<td>No default</td>
</tr>
<tr>
<td>OMIT_COMM_OPT</td>
<td>CHARACTER</td>
<td>1</td>
<td>No default</td>
</tr>
<tr>
<td>SEL_CONTROL_VAL</td>
<td>CHARACTER</td>
<td>1</td>
<td>No default</td>
</tr>
<tr>
<td>SENTENCE_KEY_CD</td>
<td>CHARACTER</td>
<td>2</td>
<td>No default</td>
</tr>
<tr>
<td>KEYWORD_SEQ_NO</td>
<td>DECIMAL</td>
<td>5,0</td>
<td>No default</td>
</tr>
<tr>
<td>TXT_SEQUENCE_NO</td>
<td>DECIMAL</td>
<td>5,0</td>
<td>No default</td>
</tr>
<tr>
<td>DIR_KEYWORD_VAL</td>
<td>CHARACTER</td>
<td>2</td>
<td>No default</td>
</tr>
<tr>
<td>DIR_KEYWORD_TXT</td>
<td>CHARACTER</td>
<td>50</td>
<td>No default</td>
</tr>
</tbody>
</table>
2) VZ3CA5.CBST129: The table structure is described in Table 4. This table has the information related to the customer account number and DIR_GR_SEQ_NO.

Table 4

**VZ3CA5.CBST129 Table Structure**

<table>
<thead>
<tr>
<th>Column_Name</th>
<th>Data_Type</th>
<th>Column_Length</th>
<th>Default_Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CUST_ACCT_NO</td>
<td>CHARACTER</td>
<td>10</td>
<td>' '</td>
</tr>
<tr>
<td>DIR_GRP_SEQ_NO</td>
<td>DECIMAL</td>
<td>5,0</td>
<td>0</td>
</tr>
<tr>
<td>KEYWORD_SEQ_NO</td>
<td>DECIMAL</td>
<td>5,0</td>
<td>0</td>
</tr>
<tr>
<td>TXT_SEQUENCE_NO</td>
<td>DECIMAL</td>
<td>5,0</td>
<td>0</td>
</tr>
<tr>
<td>DIR_KEYWORD_VAL</td>
<td>CHARACTER</td>
<td>2</td>
<td>' '</td>
</tr>
<tr>
<td>DIR_KEYWORD.TXT</td>
<td>CHARACTER</td>
<td>50</td>
<td>' '</td>
</tr>
</tbody>
</table>

3) We had joined the above mentioned two tables based on the join condition as cbst128.cust_acct_no = cbst129.cust_acct_no and cbst128.dir_grp_seq_no = cbst129.dir_grp_seq_no to form a new table which is represented as TX128129 and is located in the library CZTEMP5. The SQL query used in creation of this table was:

```sql
CREATE TABLE CZTEMP5.TX128129 AS (SELECT
A.CUST_ACCT_NO, A.DIR_GRP_SEQ_NO, A.CAP_CONTROL_NO,
A.CAP_IND_LVL_VAL, A.COMSET_SEQ_NO,
A.LST_ACCT_TYP_CD, A.LST_PUB_STAT_CD,
A.LETTER_SEQ_OPT, A.OMIT_ADDR_OPT, A.OMIT_COMM_OPT,
A.SELF_CONTROL_VAL, A.SENTENCE_KEY_CD,
B.KEYWORD_SEQ_NO,
```


B.DIR_KEYWORD_TXT FROM VZ3TX5.CBST0128 A JOIN VZ3TX5.CBST0129 B ON A.CUST_ACCT_NO = B.CUST_ACCT_NO AND A.DIR_GRP_SEQ_NO = B.DIR_GRP_SEQ_NO);

4) Similarly we had created the table FL128129, CA128129 for Florida and California states respectively in CZTEMP5 library by joining the tables VZ3FL5.CBST128, VZ3FL5.CBST129 for Florida and VZ3CA5.CBST128, VZ3CA5.CBST129 for California.

5) UNION/combined all these tables to create a new table CATXFLCOMB in CZTEMP5 library. The SQL query used to create this table was:

CREATE TABLE ZRFF.CATXFLCOMB AS (SELECT TX.* FROM ZRFF.TX128129 TX UNION SELECT FL.* FROM ZRFF.FL128129 FL UNION SELECT CA.* FROM ZRFF.CA128129 CA ORDER BY CUST_ACCT_NO, COMSET, SEQ_NO, DIR_GRP_SEQ_NO, KEYWORD_SEQ_NO, TXT_SEQUENCE_NO);

B) STEP 2:

1) The next source table which was used is CBST0014, shown in Table 5, which was present in the library VZ3CA5. This was a customer sub account table which has the customer account and sub account information. This table also had a column \texttt{dirty_dl_addr_cd} based on which we had to decide whether to pull the MAILING_ADDRESS or SERVICE_ADDRESS.
Table 5

**VZ3CA5.CBST0014 Table Structure**

<table>
<thead>
<tr>
<th>Column_Name</th>
<th>Data_Type</th>
<th>Column_Length</th>
<th>Default_Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CUST_ACCT_NO</td>
<td>CHARACTER</td>
<td>10</td>
<td>No default</td>
</tr>
<tr>
<td>CUST_SUB_ACC_NO</td>
<td>CHARACTER</td>
<td>8</td>
<td>No default</td>
</tr>
<tr>
<td>DIRY_DL_ADDR_CD</td>
<td>CHARACTER</td>
<td>1</td>
<td>No default</td>
</tr>
<tr>
<td>TYPE_OF_ACCOUNT</td>
<td>CHARACTER</td>
<td>1</td>
<td>No default</td>
</tr>
<tr>
<td>TYPE_SUB_ACCT</td>
<td>CHARACTER</td>
<td>1</td>
<td>No default</td>
</tr>
<tr>
<td>ROTARY_GPSEQ_NO</td>
<td>DECIMAL</td>
<td>5,0</td>
<td>No default</td>
</tr>
<tr>
<td>SIGNAL_START_CD</td>
<td>CHARACTER</td>
<td>4</td>
<td>No default</td>
</tr>
<tr>
<td>SIGNAL_TYPE_CD</td>
<td>CHARACTER</td>
<td>4</td>
<td>No default</td>
</tr>
<tr>
<td>LINE_TYPE_CODE</td>
<td>CHARACTER</td>
<td>1</td>
<td>No default</td>
</tr>
<tr>
<td>ACCES_REFORM_CD</td>
<td>CHARACTER</td>
<td>1</td>
<td>No default</td>
</tr>
<tr>
<td>PORT_OUT_DT</td>
<td>CHARACTER</td>
<td>8</td>
<td>No default</td>
</tr>
<tr>
<td>SUB_ACC_SVC_TYP</td>
<td>CHARACTER</td>
<td>1</td>
<td>No default</td>
</tr>
<tr>
<td>RSV_TN_IND</td>
<td>CHARACTER</td>
<td>1</td>
<td>No default</td>
</tr>
<tr>
<td>INTERLATA_RRN</td>
<td>CHARACTER</td>
<td>10</td>
<td>No default</td>
</tr>
</tbody>
</table>

2) We had pulled only the records which have the **sub account number** = ‘00000001’. The SQL query which was used to pull the data from this table is:

```
CREATE TABLE CZTEMP5.CBST14COMB AS (SELECT * FROM VZ3CA5.CBST0014 WHERE CUST_SUB_ACC_NO = '00000001')
```

C) **STEP 3**:

1) Joined the table, CZTEMP5.CA128129 which was the result of step 1, with the table CZTEMP5.CBST14COMB to form a new table which was represented as CBSSFIDD. This table was created in the library CZTEMP5.
2) The join condition used to join these tables was

\[ \text{CA128129.cust_acct_no} = \text{CBST14COMB.cust_acct_no}. \] The SQL used to create this table was:

```sql
CREATE TABLE ZRFF.CBSSFIDDI AS (SELECT
    B.DIRY_DL_ADDR_CD, A.* FROM ZRFF.CATXFLCOMB A LEFT OUTER JOIN ZRFF.CBST14COMB B ON A.CUST_ACCT_NO = B.CUST_ACCT_NO ORDER BY A.CUST_ACCT_NO,
    A.COMSET_SEQ_NO,
    A.DIR_GRP_SEQ_NO, A.KEYWORD_SEQ_NO,
    A.TXT_SEQUENCE_NO);
```

D) **STEP 4:** Built the driver query to load the target table address columns (13-17)

1) The 5 address columns in the target table IFIDDI were:
   a) IDDIAD1: address line 1
   b) IDDIAD2: address line 2
   c) IDDICTY: City
   d) IDDISAB: State
   e) IDDIZCD: Zip code

2) Joined the CBSSFIDDI with VZ3IF5.IFISB which has the address details of the customer.

3) The join condition used was **IFISB.ISBACC = CBSSFIDDI.CUST_ACCT_NO**. The SQL query which was used to
pull and load the data for these five columns is given below and this was the main driver query which was used to extract the data from the source system (CBSS):

```
SELECT
B.DIRY_DL_ADDR_CD, A.ISBACC, A.ISBMA1||A.ISBMA2||A.ISBMA3
AS MAILINGADDRESS,
A.ISBSA#||A.ISBSAN||A.ISBSAS||A.ISBSAD||A.ISBSAU||A.ISBSDA
AS SERVICE_ADDRESS, A.ISBMCT AS M_CITY, A.ISBMST AS M_STATE,
A.ISBMZP AS M_ZIPCODE, A.ISBSAC AS S_CITY,
A.ISBSST AS S_STATE, A.ISBSZP AS S_ZIPCODE FROM
VZ3CZIF3.IFISB A JOIN ZRFF.CBSSFIDDI B ON A.ISBACC =
B.CUST_ACCT_NO AND A.ISBBTN = 'P' ORDER BY
B.CUST_ACCT_NO, B.COMSET_SEQ_NO, B.DIR_GRP_SEQ_NO,
B.KEYWORD_SEQ_NO, B.TXT_SEQUENCE_NO
```

Going ahead, we had made the enhancements to this driver query for next iterations.
Figure 3. Complete mapping design of IFIDDI.

4) The flow has used three transformations in Informatica.

a) Source Qualifier transformation: In this transformation we have used the SQL override to select the columns of our choice from the source tables and is shown in the Figure 4. So from the source tables, IFISB and CBSSFIDDI, we have pulled only the columns which are used to build the address columns of the target table.
b) Router Transformation: This transformation was used to group the data based on certain conditions and is shown in Figure 5. So in our case we had grouped the incoming data based on the DIRY_DL_ADDR_CD column of the CBSSFIDDI table. If the DIRY_DL_ADDR_CD column value was 1 or NULL then marked them as one group and if the DIRY_DL_ADDR_CD column value was 2 then marked them as other group.
c) Expression Transformation: This transformation was used to write the logic for the target table columns and is shown in Figure 6. Various inbuilt informatica functions have been used to write the logic.

5) In-built informatica functions used:
   a) Concatenate: Used to join two columns. Represented by ‘||’
   b) Substr: Used to select a part of a string. Syntax is
      \texttt{Substr('XXXXXXX',start position, end position)}
   c) Ltrim(): used to trim extra spaces to the left of the string
   d) Rtrim(): used to trim extra spaces to the right of the string

E) STEP 5:

Below mentioned were the logics for the address columns of IFIDDI target table and shown in Figure 6.

For group one, i.e., DIRY_DL_ADDR_CD column value was 1 or NULL

   Concept of variable port: Complex logic can be written in the variable port and this logic can be used anywhere in the expression transformation. Created a variable port for MAILING ADDRESS with the logic –

\[
\text{Ltrim(Rtrim(ISBMA11)))||Ltrim(Rtrim(ISBMA21)))||Ltrim(Rtrim(ISBMA31))}
\]
a) IDDIAD1: Took the first 30 characters of the variable port using the substr function to populate the IDDIAD1 field. Syntax used was:

```
SUBSTR(MAILING_ADDRESS,1,30)
```

b) IDDIAD2: Took the first 31-60 characters of the variable port using the substr function to populate the IDDIAD2 field. Syntax used was:

```
SUBSTR(MAILING_ADDRESS,31,30)
```

c) IDDICTY: Trimmed all the left and right spaces of ISBMCT1 field to populate the IDDICTY field. Syntax used was:

```
LTRIM(RTRIM(ISBMCT1))
```

d) IDDISAB: Trimmed all the left and right spaces of ISBMST1 field to populate the IDDISAB field. Syntax used was:

```
LTRIM(RTRIM(ISBMST1))
```

e) IDDIZCD: Trimmed all the left and right spaces of ISBMZP1 field to populate the IDDIZCD field. Syntax used was:

```
LTRIM(RTRIM(ISBMZP1))
```

---

**Figure 6.** Group one address columns logic in expression transformation.

**COMPLETE FLOW of GROUP 1:** The complete flow of group 1 is shown Figure 7.
Figure 7. Complete workflow of group one records.

For group two i.e., DIRY_DL_ADDR_CD column value is 2 and is shown in Figure 8.

Created a variable port for BILLING_ADDRESS with the logic –
LTRIM(RTRIM(ISBSA#3))||LTRIM(RTRIM(ISBSAN3))||LTRIM(RTRIM(ISBSAS3))||LTRIM(RTRIM(ISBSSA))||LTRIM(RTRIM(ISBSDA3))

a) IDDIAD1: Took the first 30 characters of the variable port using the substr function to populate the IDDIAD1 field. Syntax used was

SUBSTR(SERVICE_ADDRESS,1,30)
b) IDDIAD2: Took the first 31-60 characters of the variable port using the substr function to populate the IDDIAD2 field. Syntax used was

```
SUBSTR(SERVICE_ADDRESS,31,30)
```

c) IDDICTY: Trimmed all the left and right spaces of ISBSAC3 field to populate the IDDICTY field. Syntax used was

```
LTRIM(RTRIM(ISBSAC3))
```

d) IDDISAB: Trimmed all the left and right spaces of ISBSST3 field to populate the IDDISAB field. Syntax used was

```
LTRIM(RTRIM(ISBSST3))
```

e) IDDIZCD: Trimmed all the left and right spaces of ISBSZP3 field to populate the IDDIZCD field. Syntax used was

```
LTRIM(RTRIM(ISBSZP3))
```

---

**Figure 8.** Group two address columns logic in expression transformation.
Complete flow for group2: Complete flow of group 2 is shown in Figure 9.

Figure 9. Complete flow of group two records.

3) **Sprint 2: 3 weeks**

A) Repeated the steps 1 through 3 of Sprint 1.

B) The logics which were used to populate the remaining columns of the target table are shown in the Table 6:
Table 6

Logic to Populate the Columns Other Than Address Fields

<table>
<thead>
<tr>
<th>Column No</th>
<th>Field Name</th>
<th>Source Field to be populated</th>
<th>Default/Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>IDDICID</td>
<td></td>
<td>Default to VZ3CZ</td>
</tr>
<tr>
<td>2</td>
<td>IDDIPST</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>IDDIEXC</td>
<td>ISBNPA</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>IDDILN#</td>
<td>ISBLIN</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>IDDICDT</td>
<td>ISBCDT</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>IDDIRES</td>
<td>ISBRES</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>IDDIDLQ</td>
<td>ISBDLQ</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>IDDIMLQ</td>
<td>--</td>
<td>Default to ‘0’</td>
</tr>
<tr>
<td>9</td>
<td>IDDIOMA</td>
<td>--</td>
<td>Leave it blank</td>
</tr>
<tr>
<td>10</td>
<td>IDDILSN</td>
<td>ISBLINM</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>IDDIFRN</td>
<td>ISBFNM</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>IDDIONF</td>
<td></td>
<td>Leave it blank</td>
</tr>
<tr>
<td>18</td>
<td>IDDIORG</td>
<td>ISBORG</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>IDDIACC</td>
<td>ISBACC</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>IDDICKT</td>
<td>ISBCKT</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>IDDICUST</td>
<td>ISBMI4</td>
<td></td>
</tr>
</tbody>
</table>

In order to populate the target columns with the above mentioned logic we had to pull additional columns from the source. This logic was same for both the type of addresses. The logics were coded in Informatica and are shown in Figure 10.

C) Enhanced the driver query of sprint 1 to populate the remaining columns of the target table. Below mentioned was the driver query for populating the target table.
SELECT
B.DIRY_DL_ADDR_CD, A.ISBNPA, A.ISBNXX, A.ISBLIN, A.ISBCDT,
A.ISBRES, A.ISBDLQ, A.ISBLNM, A.ISBFNM, A.ISBOR, A.ISBACC,
A.ISBCKT, A.ISBMI4, A.ISBDDT,
A.ISBMA1||A.ISBMA2||A.ISBMA3 AS MAILING_ADDRESS,
A.ISBSA#||A.ISBSAN||A.ISBSAS||A.ISBSAD||A.ISBSAU||A.ISBSDA AS
SERVICE_ADDRESS, A.ISBMCT AS M_CITY, A.ISBMST AS
M_STATE, A.ISBMZP AS M_ZIPCODE, A.ISBSAC AS S_CITY,
A.ISBSST AS S_STATE, A.ISBSZP AS S_ZIPCODE FROM
VZ3CZIF3.IFISB A JOIN ZRFF.CBSSFIDDI B ON A.ISBACC
  = B.CUST_ACCT_NO AND A.ISBBTN = 'P' ORDER BY
B.CUST_ACCT_NO, B.COMSET_SEQ_NO, B.DIR_GRP_SEQ_NO,
B.KEYWORD_SEQ_NO, B.TXT_SEQUENCE_NO;

Figure 10. Logic to populate the non-address columns in expression transformation.
Complete flow of sprint 2: Complete flow of sprint 2 is shown in Figure 11.

Figure 11. Complete mapping design of sprint 2.

4) **Sprint 3: 4 weeks** (IMPLEMENTATION OF SMART LOGIC)

A) After loading the address fields when we break into first 30 characters and second 30 characters, there was a problem with the address field and is explained below. In this example, yellow are characters 1-30, red are 31-xx, Allen is city, TX is state, 75002 is zip:

6679 Southwest Central Expressway  Allen TX 75002.

B) If we parse solely on 1-30, and 31-60, it would break out like this:

6679 North Southwest Central Expressway  Allen TX 75002
C) We want it to break like this. So that the complete word is together.

6679 North Southwest Central Expressway Allen TX 75002

D) So to avoid this problem, below mentioned logic shown in Figure 12 and Figure 13 for both the mailing address and service address respectively was implemented.

For Mailing address,

\[
\begin{align*}
\text{IDDIAD1} &= \text{ltrim(rtrim(ISBMA11))} \\
\text{IDDIAD2} &= \text{ltrim(rtrim(ISBMA21))} || \text{ltrim(rtrim(ISBMA31))}
\end{align*}
\]

Figure 12. Smart logic for mailing_address.

For Service Address,

\[
\begin{align*}
\text{IDDIAD1} &= \text{ltrim(rtrim(ISBSA#3))} || \text{ltrim(rtrim(ISBSAN3))} || \text{ltrim(rtrim(ISBSA S3))} \\
\text{IDDIAD2} &= \text{ltrim(rtrim(ISBSAD3))} || \text{ltrim(rtrim(ISBSAU3))} || \text{ltrim(rtrim(ISBSD A3))}
\end{align*}
\]

Figure 13. Smart logic for service_address.

5) **Sprint 4: 4 weeks** (Pull the service address from IFISM table).

A) Till then, both the service address and mailing address were being pulled from IFISB table. But there was a change in requirement to pull the service address from the IFISM table since IFISB no longer has the
service address and the Informatica coding is shown in Figure 14.

Below Query was used to pull the service address:

SELECT
B.DIRY_DL_ADDR_CD, A.ISBACC, A.ISBMA1||A.ISBMA2||A.ISBMA3
as Mailing_Address,
g.ISMST#||g.ISMSTR||g.ISMSST||g.ISMSDR||g.ISMSSD||g.ISMSDA as Service_Address, A.ISBMCT as M_CITY, A.ISBMST as M_STATE,
A.ISBMZP as M_ZIPCODE, A.ISMCTY as S_CITY, A.ISMSTA as S_STATE, A.ISMZCD as S_ZIPCODE FROM vz3czif3.ifisb a join
zrff.cbssfiddi b on a.isbacc = b.cust_acct_no join vz3czif3.ifism g on
a.isbcn#=g.ismcn# and a.isbbtn = 'P' ORDER BY B.CUST_ACCT_NO,
b.COMSET_SEQ_NO , b.DIR_GRP_SEQ_NO,
b.KEYWORD_SEQ_NO, b.TXT_SEQUENCE_NO;
Figure 14. Address fields populated from IFISM table.

B) The next step was to create a session for this mapping in the work flow manager. After that we had ran that session to populate the data. This is shown in Figure 15.
6) **Deployment Locations.**

- The development work had first been developed in developer environment and the developer had unit tested their code.
- After unit test was done, the code will has been migrated to the test environment for the testers to test.
- Once the code was tested by testers, the final code has been deployed into production environment.

**Data Analysis**

Can SQL be used as analytics tool? The answer is yes. Structured Query Language is a custom-built language which is used for working with relational databases in which data is arranged in rows and columns. There are different inbuilt
functions in SQL to retrieve the data and some of the functions which were used in our analysis are:

- **Count (**\(^*\)**): This function will count the number of rows from the entire table.
- **Concatenate**: This function will join two columns.

Data analysis was performed using SQL analysis and excel sheet. The data which was retrieved from the table was exported to excel sheet.

**Budget**

The cost allocated to the project was perfectly utilized and the whole project was completed with the allocated costs. No additional costs were incurred at the time of development and successful completion of the project.

**Timeline**

The Planning phase was completed by end of November and the sprint 1 was started in December and was progressed for 6 weeks until mid-January. After sprint 1 sprint 2 was started by mid-January and was progressed until January end. Sprint 3 was completed by February end and sprint 4 was completed by March end. The code was moved to production on April 1\(^{st}\) and is shown in Table 7.
Table 7: Timeline of the project

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Planning phase</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Iteration 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Capstone Proposal</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Iteration 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Iteration 3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Iteration 4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Go Live &amp; Support</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Final Defense</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- On Site Project Tasks
- Capstone project Tasks
Chapter 4: Data Presentation and Analysis

Introduction

This chapter will primarily focus on data presentation and data analysis. Data was represented in the form of relational tables which have rows and columns. Data analysis was done using the SQL analysis method.

Data Presentation

After code was migrated to production environment and running the jobs, the target table was loaded with the data and is shown below. All the default values are populated as expected. All the address fields were also populated as expected and is shown in Figure 16. Data was viewed in the target table by using the below mentioned query: Select * from vz3czif5.ifiddi.

Figure 16. Target data address fields in VZ3CZIF5.IFIDDI table.
Data Analysis

As discussed in the previous chapter data analysis was done using the SQL queries. As per the requirement the address of the CBSS customers should be loaded into the target table. So the total number of CBSS customers who address should be populated is shown in figure 17.

```sql
SELECT COUNT (*) FROM (SELECT B.DIRY_DL_ADDR_CD,
A.WASBACC, A.WASBMA1||A.WASBMA2||A.WASBMA3 as Mailing_Address,
g.WASMST#||g.WASMSTR||g.WASMSST||g.WASMSDR||g.WASMSSD||g.WA SMSDA as Service_Address, A.WASBMCT as M_CITY, A.WASBMST as M_STATE, A.WASBMZP as M_ZIPCODE, A.WASMCTY as S_CITY,
A.WASMSTA as S_STATE, A.WASMZCD as S_ZIPCODE FROM vz3czif3.if wasb a join zrff.cbssfiddi b ON a.wasbacc = b.cust_acct_no join vz3czif3.if wasm g on a.wasbcn#=g.wasmcn# and a.wasbbtn = 'P' ORDER BY B.CUST_ACCT_NO, b.COMSET_SEQ_NO, b.DIR_GRP_SEQ_NO,
b.KEYWORD_SEQ_NO, b.TXT_SEQUENCE_NO) t; --1607945
Figure 17. Total number of records fetched from source.

The total number of records that were loaded into the target table (shown in Figure 18) was found out by using the count (*) query on the target table which is

```
SELECT COUNT (*) FROM VZ3IF5.IFIDDI; -- 1607945
```
Figure 18. Total number of records loaded into the target table.

The total number of records which were extracted from the source and the total number of records which were loaded into the target are matching which indicated that the data which was extracted from the source was loaded into the target without any mismatch.

Once the address fields were populated then the reports were generated using the export functionality in the Oracle SQL developer. Two reports were generated and excel sheets were used to report the data:

1) Records which have addresses: Shown in figure 19. This report was generated by using the SQL query:

```sql
SELECT * FROM VZ3CZIF4.IFIDI WHERE IDDIORG LIKE '%CBSS%' AND IDDIAD1 <> '' AND IDDIAD2 <> '';
```
Figure 19. Total number of records which have address fields populated.

2) Records which don’t have the addresses: shown in Figure 20. The report was generated using the SQL query:

```sql
select * from vz3czif4.ifiddi where iddiorg like '%CBSS%' and IDDIAD1 = IDDIAD2;
```

A total of only five records were present which does not have the address fields populated.
Figure 20. Total number of records for which address fields are not populated.

The records for which we don’t have the address were updated manually and were loaded into the AS400 DPI tables. So these addresses were used to send the directories to the customers.
Chapter 5: Results, Conclusion, and Recommendations

Introduction

This chapter mainly focuses on the results that are achieved after the successful implementation of the project, conclusion and the recommendation to enhance the output of the project.

Results

Agile methodology has been successfully implemented in the conversion of data using the Informatica ETL tool. The data which was loaded into the target database was accurate and was as expected.

Project Questions

1. *What were the improvements that can be made by using this approach?*

   Waterfall Model: Previously the same project has been implemented to load the target database using waterfall approach. The total time that was taken to complete the project was a period of 12 months.

   Agile Methodology: The total time which was taken to complete the project was 6 months. This implies that the overall reduction in time was cut down to 50%.

2. *What was the overall impact at the organization level by following this approach?*

   Cost plays a major role in success of any industry or company. By implementation of agile methodology the time which was invested was decreased by 50%. This was directly proportional to cost reduction.
Waterfall Model: $25/hr x 2080 hrs (for 53 weeks) = $52000

Agile methodology: $25/hr x 1080 (for 27 weeks) = $27000

Overall cost was reduced by $25000.

3. What were the target deliverables that are achieved by the implementing the agile approach?

The final data which was extracted from the source was successfully loaded with the customer data and their addresses. These address fields are used to send the information about the directory book, promotional packs to the customers.

Conclusion

As the project main goal was to implement the agile methodology to save time, cost and have flexibility to incorporate the change requirements, client was very much satisfied with the outcome and they were very much interested in implementing the agile methodology to other projects as well. They were very much impressed by the way requirement changes have been implemented using the agile methodology. Client was also satisfied by the amount of time and money saved by the implementation of agile methodology.

Recommendations

Many reporting tools are available in market like SAP BO, cognos, microstrategy which can be better used in generation of reports.

Push down optimization can be effectively implemented if time and resources were more.
References


