HÖGSKOLAN
I GÄVLE

FACULTY OF ENGINEERING AND SUSTAINABLE DEVELOPMENT
Department of Industrial Development, IT and Land Management

System Information Model suitable for Application Management Organization

Izzet Canciger

2013

Diploma Work, University diploma, 15 HE
Computer science
Internet Technology 120 HE

Examiner: Carina Pettersson
Supervisor: Bengt Ösberg

Preface

I want take this opportunity to thank everyone at Sandvik IT for all their help and friendly welcome.

Thanks to Kjell Persson at Sandvik IT for giving me the opportunity to perform this thesis.

A special thanks to my supervisors on Sandvik IT, Christer Olsson and Peter Brodin for your invaluable help.

And thanks to my supervisor at University of Gävle, Bengt Östberg for your guidance.

Finally, I would like thank my wife and daughter for their patience and understanding.

Gävle, May 2013
Izzet Canciger

System Information Model suitable for Application Management Organization

by

Izzet Canciger

Academy for Technology and Environment
University of Gävle

S-801 76 Gävle, Sweden

Email:
cancigerizzet@hotmail.com

Abstract

The Delivery Organization *Windows Development* at *Sandvik IT GSS* is today responsible for managing over 100 different systems internally and globally, some acquired but most proprietary. The systems have Life Cycle and dependencies of other applications, techniques etc. and Service Level Agreement (SLA) with customers. The details for these systems are now scattered in a variety of locations such as web pages, spreadsheets, word documents, Notes databases and by knowledge of the employees. Today they are unable to get a clear view of the systems; the applications and techniques depend, how long life cycle each have etc. The project consist of creating a database model by gathering information needed, analyze it and develop an information model and if time allows develop a simple application to visualize the concept. With this work, the information will be gathered in one place and the employees will easier get a clear view of the systems and their dependencies, offer a higher level of service and will be more effective in their work.

Keywords: System Information Model, Systems Design, Information Model, Database Model, Business Analysis, Application Management.

CONTENTS

1 Introduction

1.1 Company description

Sandvik AB contains of 5 business areas [1]
To learn more about each business area follow the links.

Sandvik Construction [2]
"This business area provides solutions for virtually any construction industry application encompassing such diverse businesses as surface rock quarrying, tunneling, excavation, demolition, road building, recycling and civil engineering. The range of products includes rock tools, drilling rigs, breakers, bulk-materials handling and crushing and screening machinery."

Sandvik Machining Solutions (SMS) [3]
"Manufacturer of tools and tooling systems for advanced industrial metal cutting. Products are manufactured in cemented carbide and other hard materials such as diamond, cubic boron nitride and special ceramics."

Sandvik Materials Technology (SMT) [4]
"Manufacturer of high value-added products in advanced stainless steels, special alloys, as well as metallic and ceramic resistance materials for the most demanding industries. The product areas comprise Tube, Strip, Primary Products, Wire and Heating Technology."

Sandvik Mining [5]
"Supplier of equipment and tools, service and technical solutions for the mining industry. The offering covers rock drilling, rock cutting, rock crushing, loading and hauling and materials handling."

Sandvik Venture – Research and development (R&D) [6] [7]
"They are aiming to create the best possible environment for growth and profitability in attractive and fast-growing operations. The product areas comprise Sandvik Hard Materials, Diamond Innovations, Wolfram and Sandvik Process Systems."

1.2 Sandvik IT [8]

Sandvik IT [9] is wide and divided business area, where every area has its own IT and there is areas besides the local IT.
Because of the complexity of *Sandvik IT*, I will focus on *Windows Development* the group where I will do this project.
The red thread through Sandvik IT organization to Windows Development is: Sandvik IT – GSS – Standard Services – Application Services – Window Development. *See figure 1*.

Facts
Sandvik IT consist of CIO, Communication, CIO Office, HR (Human Resources), BIO SMS, BIO Mining/Construction, BIO SMT, BIO Group Functions *and GSS (Global Shared Services*.

IT Global Shared Services (GSS) consist of Head of GSS, HR, Process Support & Control, IT – Finance, Delivery Mining/Construction, Delivery SMT, Delivery SMS, Delivery Group & Venture, Business Services and *Standard Services*.

Standard Services consist of Head of Standard Services, Security, HRA, Operational Development, *Application Services*, Infrastructure Services, Support Services and Integration Services.

Application Services consist of Manager, *Windows Development*, Mainframe Development, Notes Development and SharePoint Development. See figure 1.

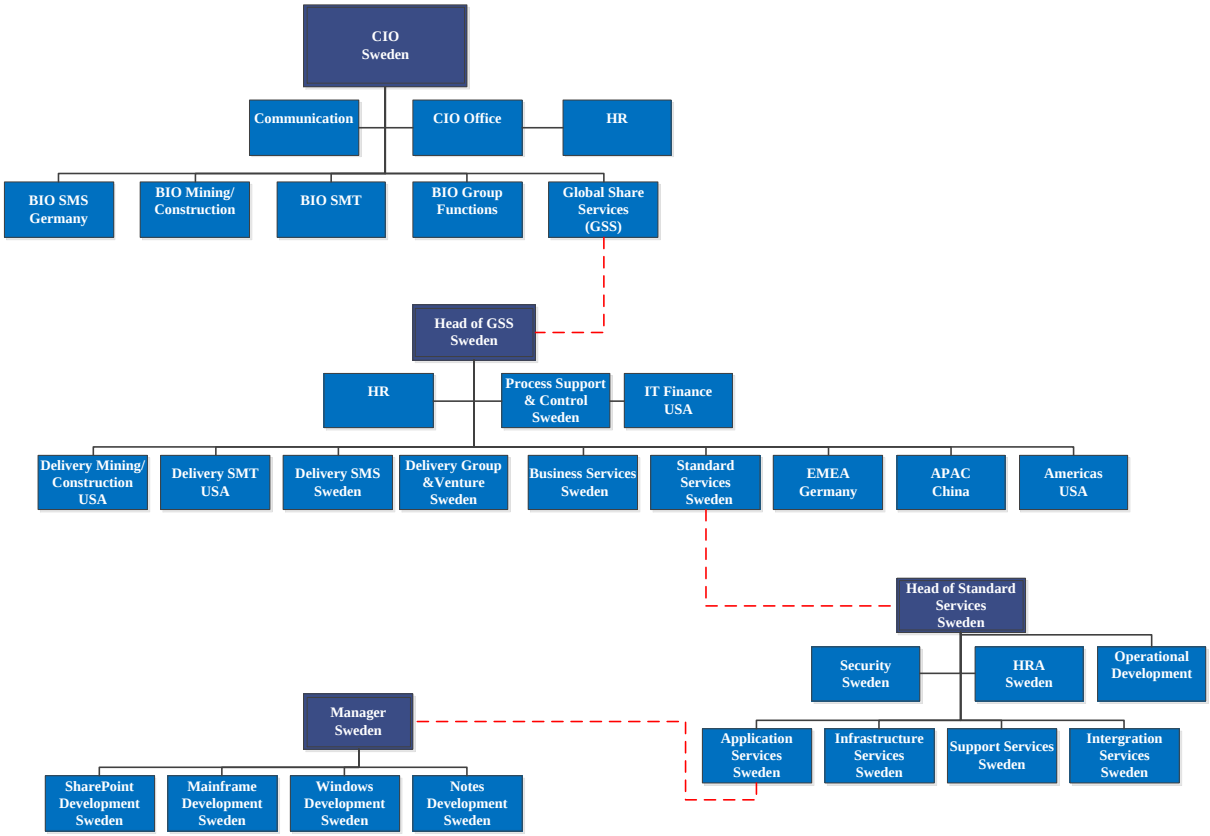


Figure 1: Red thread to Windows Development

2 Background

The delivery organization *Windows Developments* at *Sandvik IT GSS* are today responsible for managing over 100 different systems some acquired but most is house developed. The systems range from 15 years old to new system still under development.
The complexities of the systems vary from simple "stand alone" to complex systems with many different technologies and integrations.
Example of details: Use of technology, which versions of components integration techniques used, life cycle of programs/applications and the database linkages.

2.1 Problem

Windows Development currently has an inefficient system for managing the information of the systems. The information about the systems is today scattered in many places, such as web pages, spreadsheets, Word documents, Notes databases and by knowledge of the employee. Most of the information is out of date and hard to maintain. This is very problematic. Some customers have an agreement called SLA (Service Level of Agreements) where life cycle advisory is a part of the SLA. Today it is hard to live up to this part of SLA.

2.2 Purpose

The main goal is to gather information about the systems life cycle in one place.
My task with this thesis is to create architecture for a *life cycle system* suitable for holding the information needed by an application management organization. This will be done by gathering information needed, analyzing it, develop an information model and then create a database model. If time allows develop a "Proof of concept".
With this work, the information will be gathered in one place; it will be easier to a get a clear view of the systems and their dependencies, offer better service, be more effective in their work, be able to deliver more efficient and be able to act more proactively.

3 Methods

According to Valacich *et.al* (2001) System Development Life Cycle (SDLC) there are 4 phases:
Phase 1: Systems Planning and Selection. There are two primary activities.
The first task is to identify the need for a new or improved system, analyze, prioritize and translate it. Creates a report with development schedule and sends to responsible department. *The second task* is to investigate and produces a detailed plan with budget for the project. Then decide whether the project is feasible or not. When the two activities are finished a final presentation is performed for the organization management.
Phase 2: System Analysis which has three steps. *In the first step* makes the analyst a careful analysis of the operations and their current systems. Determine with the users what they want with the system. *The second step* is to study the requirements and

structure them by creating a logical design. The third *step* is to create an alternative design and compare this according to requirements, cost, labor and technical levels.
Phase 3: System Design. Here the analyst will convert the logical design to a physical design. Decide in what language, which database systems, which hardware platform, what kind of file structure, which operating system and network environment that will be used for the system.
Phase 4: Systems Implementation an Operation.

3.1 Business Analysis

Business Analysis is a method to identify how the process of work is today and to see the needs, requirements or conditions to determine a solution. This analysis is critical for the success of system development.

3.2 Workshop

The purpose of the workshops is to gather the information elements needed by the different areas of the department.

3.3 Information Model

An Information Model is an organizational framework to categorize the information for the *Information System Model* and to get a clear picture of concepts and relationships for the *Information System Model*. To see what kind of *attributes*, *values* and *relationship* it would contain.

3.4 Database Model

The Database Model is a logical plan for the *life cycle system* based on Information Model.

The database model for the *lifecycle system* is a relation based database model. [10]
A relational database is a database that spreads across several tables. Each table is made up of records up of records and fields in which records (rows in a table) represents a collection of fields (columns) affecting the same objects and fields represent data belonging to multiple items (such as list of telephone numbers)

Basic Concepts:

- Tables are also called register
- Record i.e. information that belong together and occupy a row in a table such as personal data, address, phone number, e-mail address.
- Field – a specific part of the information contained in a record. Field information is included in a table´s column, such as the field of phone numbers.
- Field name – The name that describes column content.
- Field length – maximum number of characters a field may contain.

Each table in a relational database is identified by a unique name and the name is used by the database to find the table and data that lies therein.

In a relational database, data is manipulated by performing an operation on the table´s records of fields. If you want for example get a list of all persons with surname "Smith" from the table "Students" you search in the table "Students" Field "LastName" for the data that meet your search criteria.

3.5 Proof of Concept

Proof of Concept is a simple realization of an idea whose purpose is to verify that the theory is capable of development in a useful way, also called prototype.

4 Realization

In this report I have developed the architecture for the *life cycle system*. I have conducted the Phase 2 and 3 according to SDLC. My approaches to this task were by meeting the client once a week and discuss the work performed since the previous meeting. We summed up this and discussed what should be done until the next meeting. Depending on the task, the time between the meetings varied.
In order to produce a *life cycle system* these steps are necessary, performing a Business Analysis, conduct a Workshop, create an Information Model, create a Database Model and finally develop a Proof of Concept (prototype).
In my case I will conduct a system analysis (Phase2, first step) through Business Analysis and a Workshop.
In the second step my logical design is an Information Model.
The third step in Phase 2 was done iterative. The economic aspect was not included in my work.

4.1 Business Analysis

Business Analysis was conducted to gain an understanding of the process, through interviews with coordinators on Windows Development and another interview with the Solution Architect and a Developer.
To the interview I had prepared some questions:

- What does *Windows Development* work with?
- How are *Windows Development* organized?
- What is your main task?
- How do you work?
- What is necessary for you work?
- Where is the information about your systems today?
- What are the difference between and application and a system?

4.2 Workshop

A workshop was held to highlight what information should be included and the purpose of the information.
The workshop was carried out by brainstorming on post-it notes. Participants had 20 minutes to write down the contents they wanted and why they needed it.
When 20 minutes had passed they got up one by one and stacked without any discussion the notes up on a whiteboard divided by **contents** and **purpose**. After that they had 10 minutes discussion to sort out similar **contents** and **purpose** and to add more. But 10 minutes became 20 minutes because of an intensive discussion.
Closure for the workshop was the priority. We focused only on the **purpose**, but by prioritizing **purpose** the **content** was automatically prioritized.

4.3 Information Model

Create an information model for the *lifecycle system* by analyzing the business analysis and the workshop. An example question we need to answer:
How and when does another life cycle affect our application´s life cycles? For example: which systems are dependent of Windows 2003 Server.

We created two information models to see and compare which information model that was best suited for the *lifecycle system.*

4.4 Database Model

Through the information model build up the relations and data and create a database model.

4.5 Proof of Concept

A Proof of Concept will be created after having submitting this thesis report. A simple application of the *lifecycle system* will be made to visualize dependencies between the systems. This will be developed in C#. An application will be created using SQL Server 2008 without history tables. All data will be entered manually in the database.
The Proof of Concept will be able to produce one/several reports in the following order.

- System report with the following content: *See Appendix B*
  - An indicator for any conflicts between the systems lifecycle and the systems components lifecycle
  - End Of Life of the System
  - List of applications it has
  - Application´s End Of Life
  - List of Techniques it uses
  - Technique´s End Of Life

- Technique Report with the following content: *See Appendix C*
  - Technique Name
  - End Of Life of the Technique

- List of Systems that are using the Technique
- Systems End Of Life

- Status list with the following content: *See Appendix D*
  - An indicator for any conflicts between the systems lifecycle and the systems components lifecycle
  - A list of Systems

- Application Report with the following content: *See Appendix E*
  - Application Name
  - A list of Data Carriers

5 Results

5.1 Business Analysis

These are the answers to the questions (title 4.1) and the result of the interview with the coordinators.

Q: What does Windows Development work with?
Answer: Management and development of applications.

Q: What is necessary for your work?
Answer: The most important thing is the *life cycle* of all technical needs/dependence systems we work with have.

Q: Where is the information about you systems today?
Answer: The information is contained in web pages, spreadsheets, word documents, Notes databases and by knowledge of the employee.

Q: What is the difference between an application and a system?
Answer: A system can consist of one or more applications and an application is an application

Q: How is Windows Development organized?
What is your main task?
Answer: *Windows Development* is divided in to 4 divisions with 1 Functional Manager and 28 employees.
Organization and division are described in the following tables. There is also a tester for all divisions.

Windows Group:

Amount managed applications	**About 100 applications**
Staff	1 Team coordinator 1 Solution Architect 9 Developers (1 in India)

Main task	Development
Support	Rolling schedule
Work form	Line form

Logistic Group

Amount managed applications	7 applications
Staff	1 Team coordinator 1 Solution Architect 11 Developers(2 in China and 2 in USA)
Main task	Development
Work form	Project form

Star – system

Staff	1 AM Manager

Tekla – system

Staff	1 AM Manager

Internal systems

Staff	1 AM Manager

Q: *How do you work?*
Answer: *Windows Development* works with *case management* (development, incident, request, order etc.). Where cases reports in *Mail, project portal (SharePoint solution) or phone.*

*Case management method**:***
Windows Group manages cases by KANBAN [11] – a Japanese scheduling system. *Kanban* literally means signboard or billboard. The idea is to visualize cases by splitting the work into pieces, write each item on a post it note and put it on a wall. On the wall there are named columns to illustrate where each item is in the workflow. The *Windows Group* has following columns and demands: *see figure 2*

- Incoming
  - Update Windows AM Maintenance with your name
  - Update note with <start date>
- Prepare
  - Document the "demands"
  - Task posted in TFS(Team Foundation Server)
- Develop
  - Test case are written
  - Claims are unit tested
  - System documents are updated
- Internal Acceptance Test
  - Claims are tested according to Test case
  - Free tested/Repression tested
  - Free test are documented
- Customer Acceptance Test

    - OK from customer in writing and Layout is in the AM Maintenance
    - Timing of production setting agreed
- Deploy
    - Confirm with customer that everything works
    - Update note with <end date>
- Done

If there is a stop with a case, they take a new. Each morning they have meetings at the board where they plan how to they should proceed with cases. With post-it notes they organize and keep track of the cases. *See figure 2*.

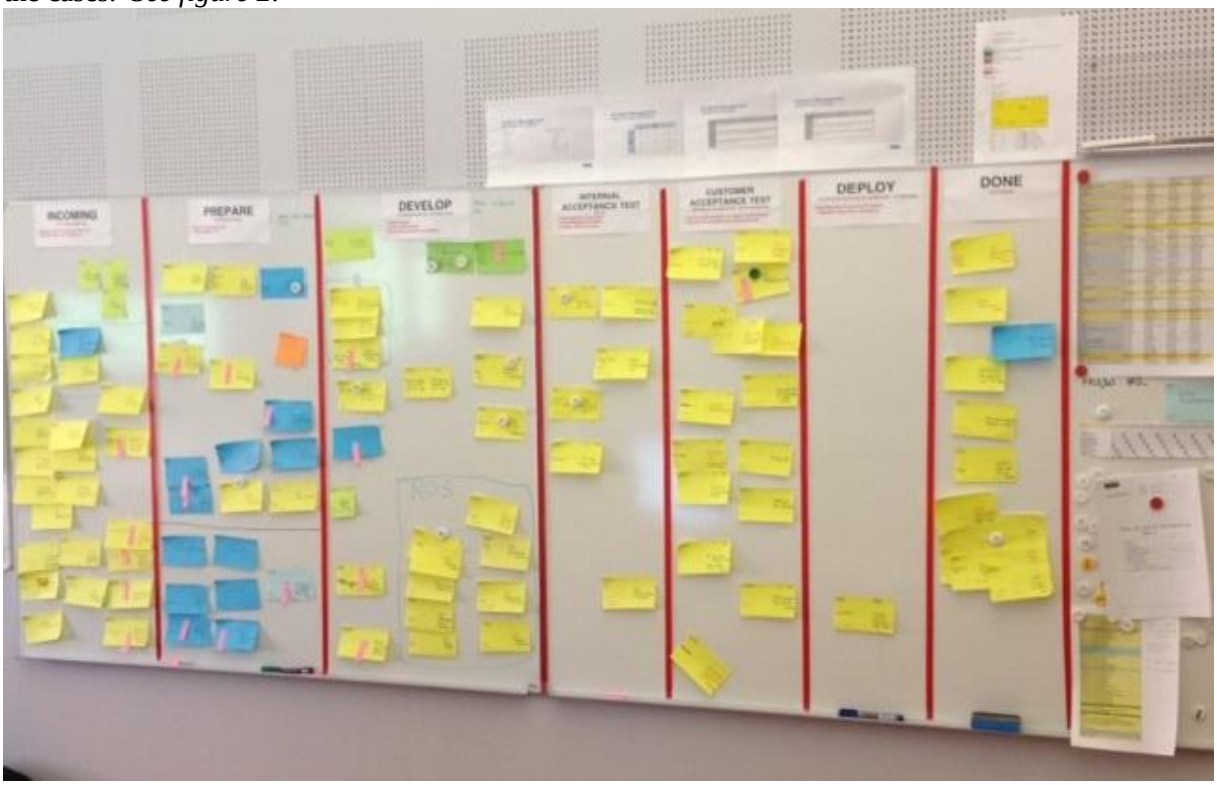


Figure 2: KANBAN Board

Logistic Group manages cases by the *Scrum* [12] – method. Time period 2-4 weeks,
Scrum – The workflow is similar *Kanban* but it has more direct contact with the customer and there is more coordination and planning, also great for projects.
Scrum is based on *Sprint* – "A sprint is the basic unit of development in Scrum. The sprint is a "timeboxed" effort, i.e. it is restricted to a specific duration. The duration is fixed in advance for each sprint and is normally between one week and one month."
Logistic Groups Sequence: *Sprint* comes in and they have 2-4 weeks, 2/3 is development time and 1/3 is for testing. If there is a new task before the first is finished this will be implemented in the next *Sprint*.
How you plan for each *sprint. See figure 3& 4.*

- First *prio meeting*, planning with customer.
- Estimate available man hour.
- Check the task list and calculate how many tasks that can be done.
- In the end *sprint* perform test.

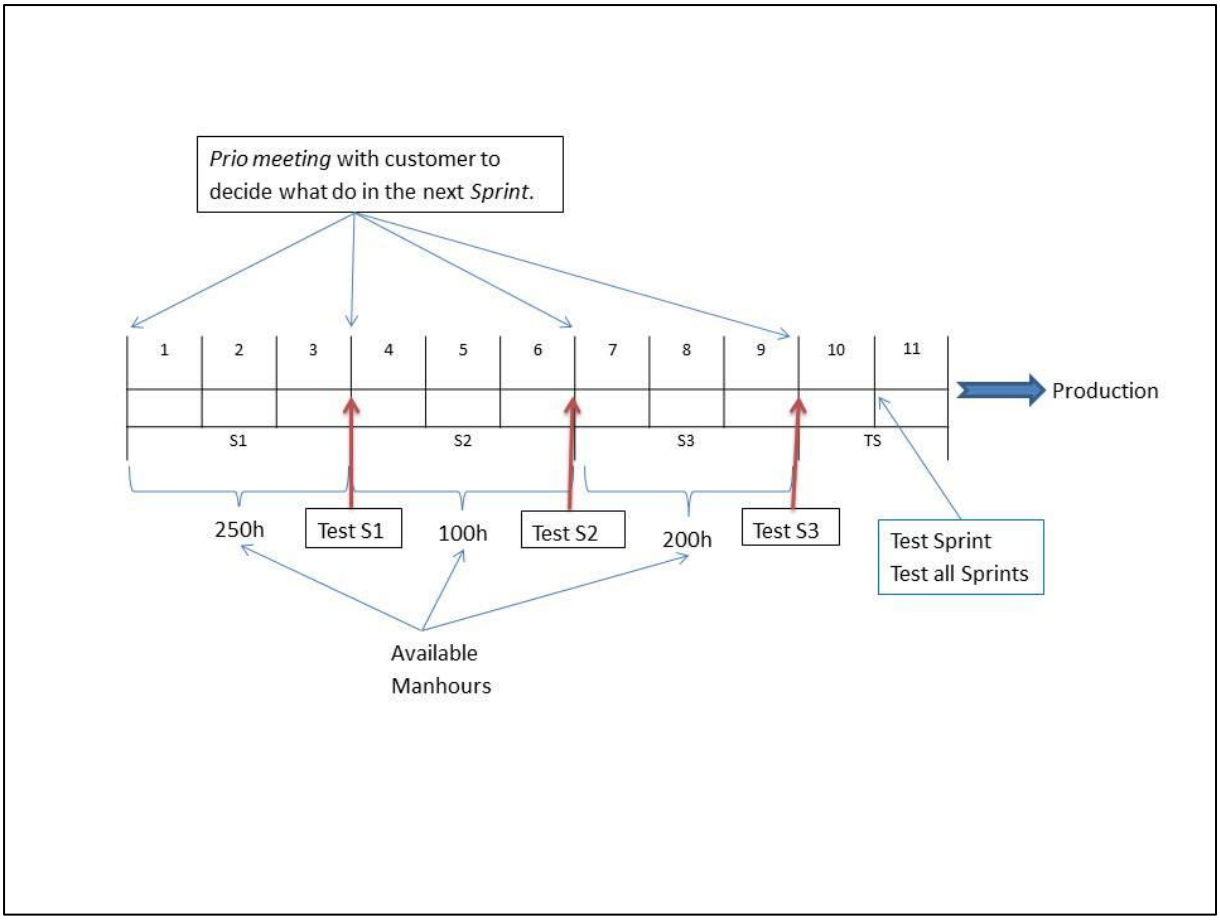


Figure 3: Sprint Chart

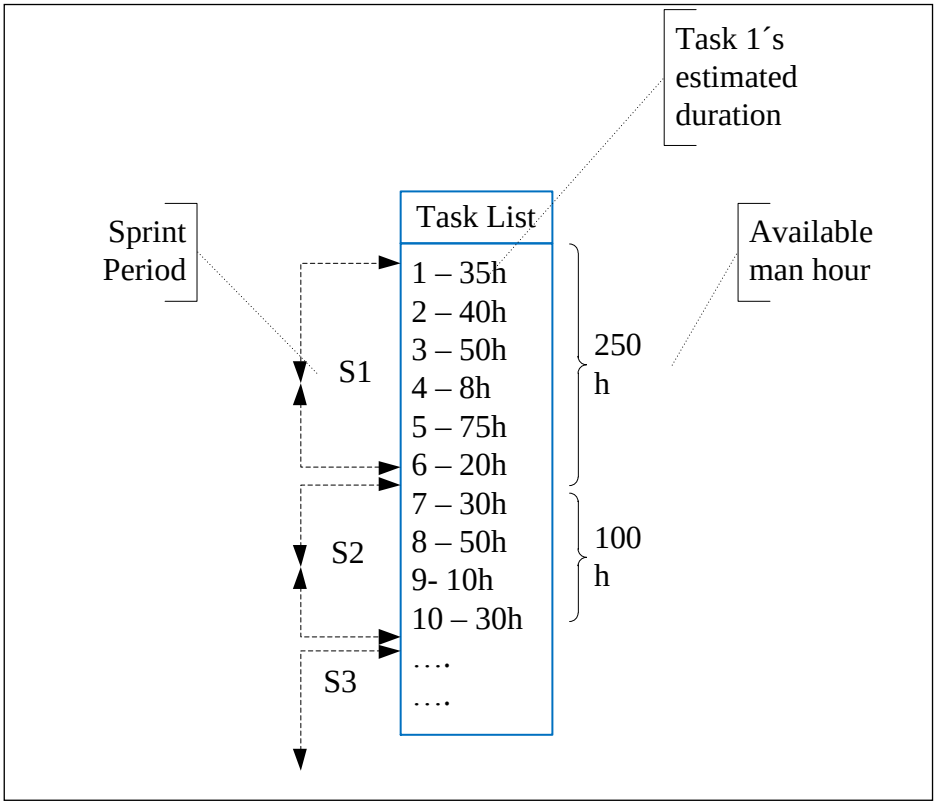


Figure 4: Scrum Task List

5.2 Workshop

Through the workshop, we found out which information should be included in the database as well as the purpose of the information. The most important information is: *See Figure 5*

- The Life Cycle information for the Systems, Techniques and Applications.
- To be able to filter *Systems* by *Techniques*.
- See what *Systems* are dependent on one technology generation.
- Easy administration such as Updating and Editing.

Participants: Åsa Aspfors *AM Coordinator*. Peter Brodin, *Developer* and Christer Olsson, *Solution Architect*.
Workshop leader: Izzet Canciger

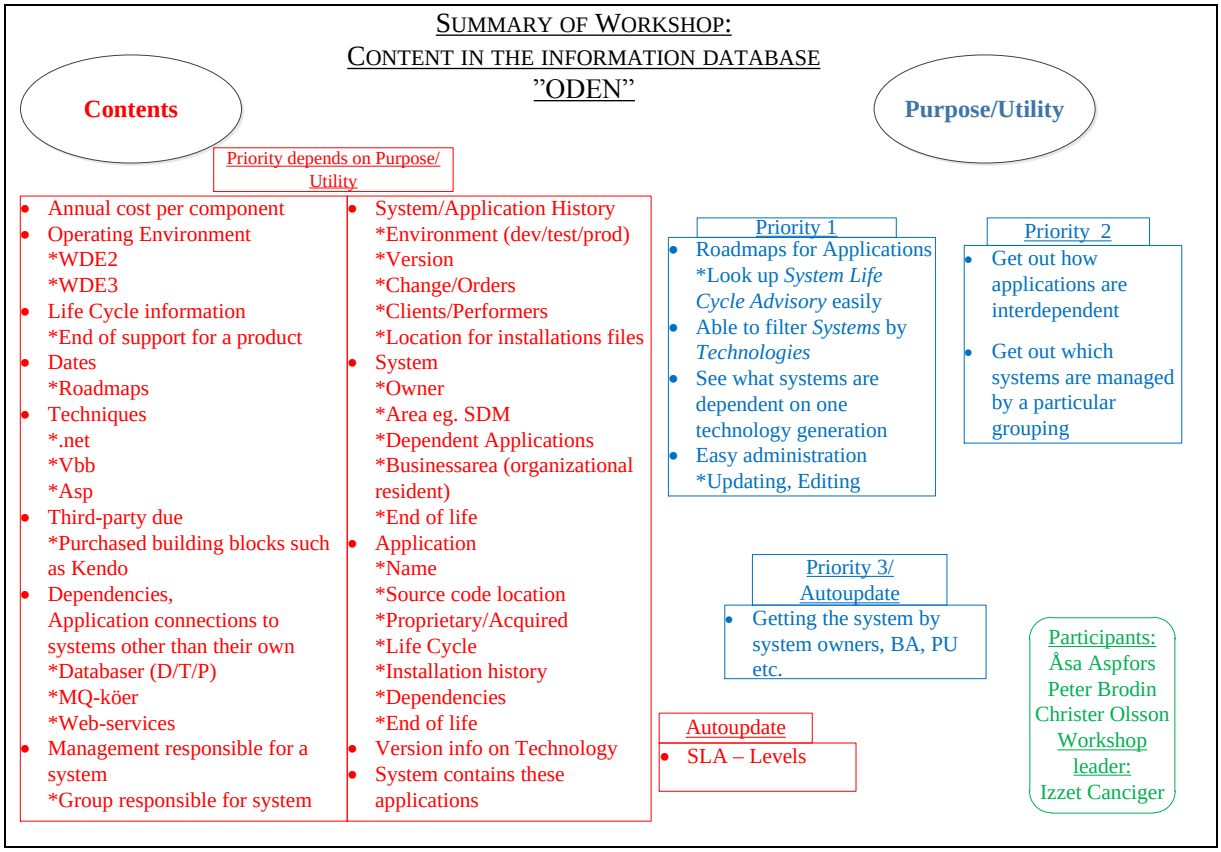


Figure 5: Summary of Workshop

5.3 Information Model

The resulting information model has seven entities with attributes (*see Appendix F*) and relations.

- System,
- Application
- Life Cycle
- Technique

- Installation History
- Operating Environment
- Integration (first information model *see figure 6*) Data Carrier (second information model *see figure 7*)

We made two different models that we compared. We choose the second information model. W come concluded that Data Carrier also uses Technique. *See figure 6 & 7*
The difference between the first information model and the second information model are the relations to Technique. In the first information model Technique is linked directly to Application and in the second information model it is linked both Application and Data Carrier/Integration. We choose the second solution because it was considered more suitable for the needs of *Windows Development*.

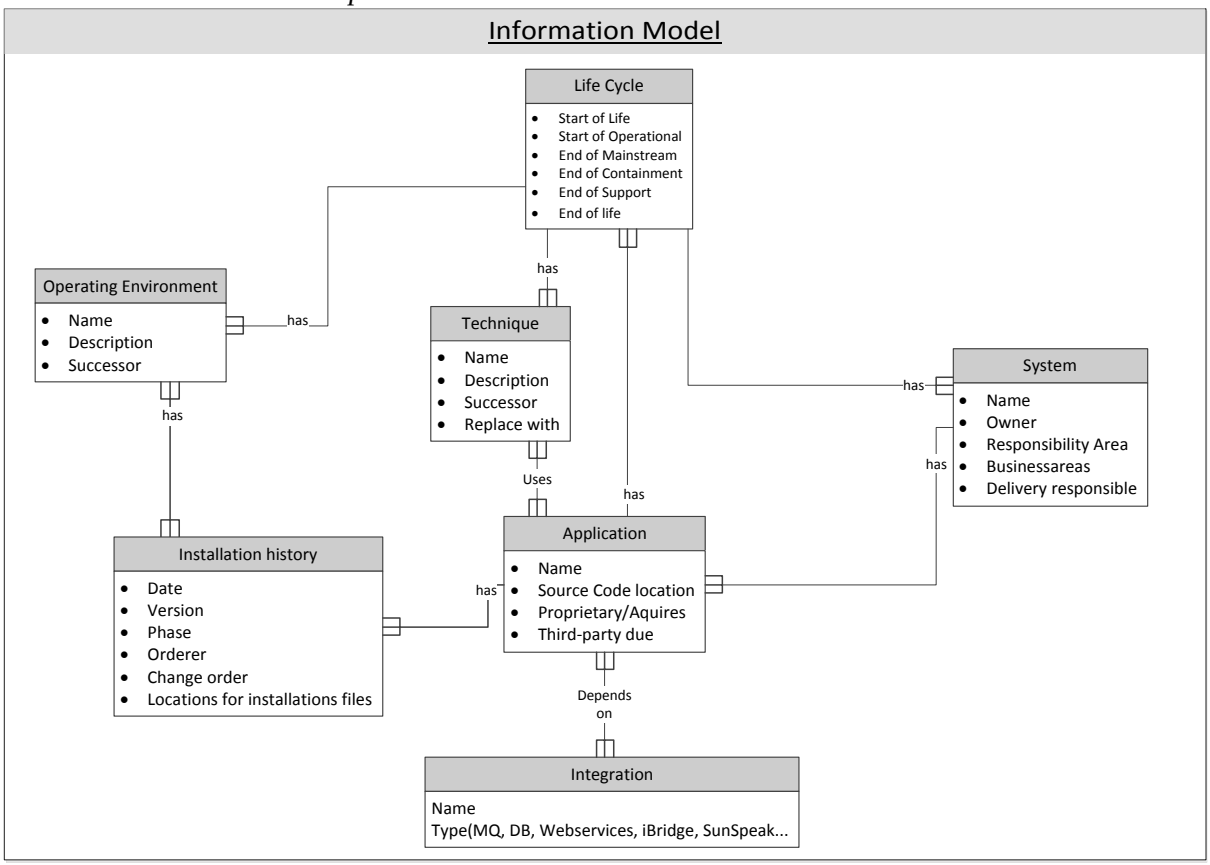


Figure 6: First Information Model

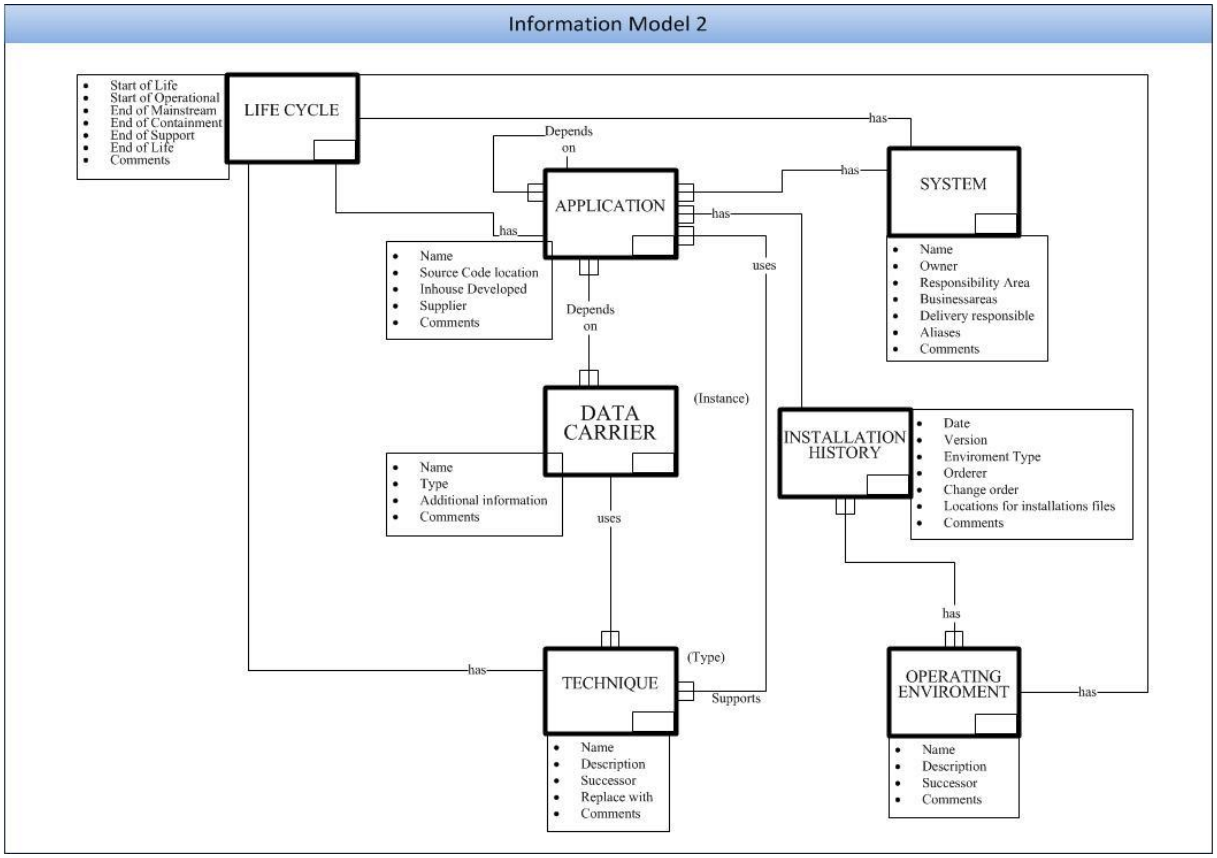


Figure 7: Second Information Model

5.4 Database Model

The Database Model was created from the information model. In the first layout the seven entities with the attributes were transformed into seven tables, move some attributes to sub tables. I added a unique ID to each table. *See figure 8.* After several meetings and several layouts with the Solution Architect and the Developer we had a result. We concluded that we needed seven main tables, twelve sub tables and two relational tables.

The hardest case was how to conduct a history/recording of the changes. We had two main solutions. The first was so called *Instance* solution, where every change was saved as Instance. Each table except Person had these attributes: *see figure 9 and Appendix G.*

- InstanceID – Unique ID with automatic serial number
- Person – the person who made change
- Timestamp – date and time when change was made
- Delete – if the System is delete(end of life)

The second solution is a *shadow* solution where there is a shadow table for all tables. The shadow tables are used for saving the history. For this to be possible the junction tables must have a unique ID, *ApplicationTechniqueID* and *ApplicationDataCarrierID* to communicate with his shadow. The shadows have additional column for storing the value of the primary key of the main tables.

Two new attributes was added *ChangeBy* and *Timestamp* in each table for tracking. *See figure 10 and Appendix G.*

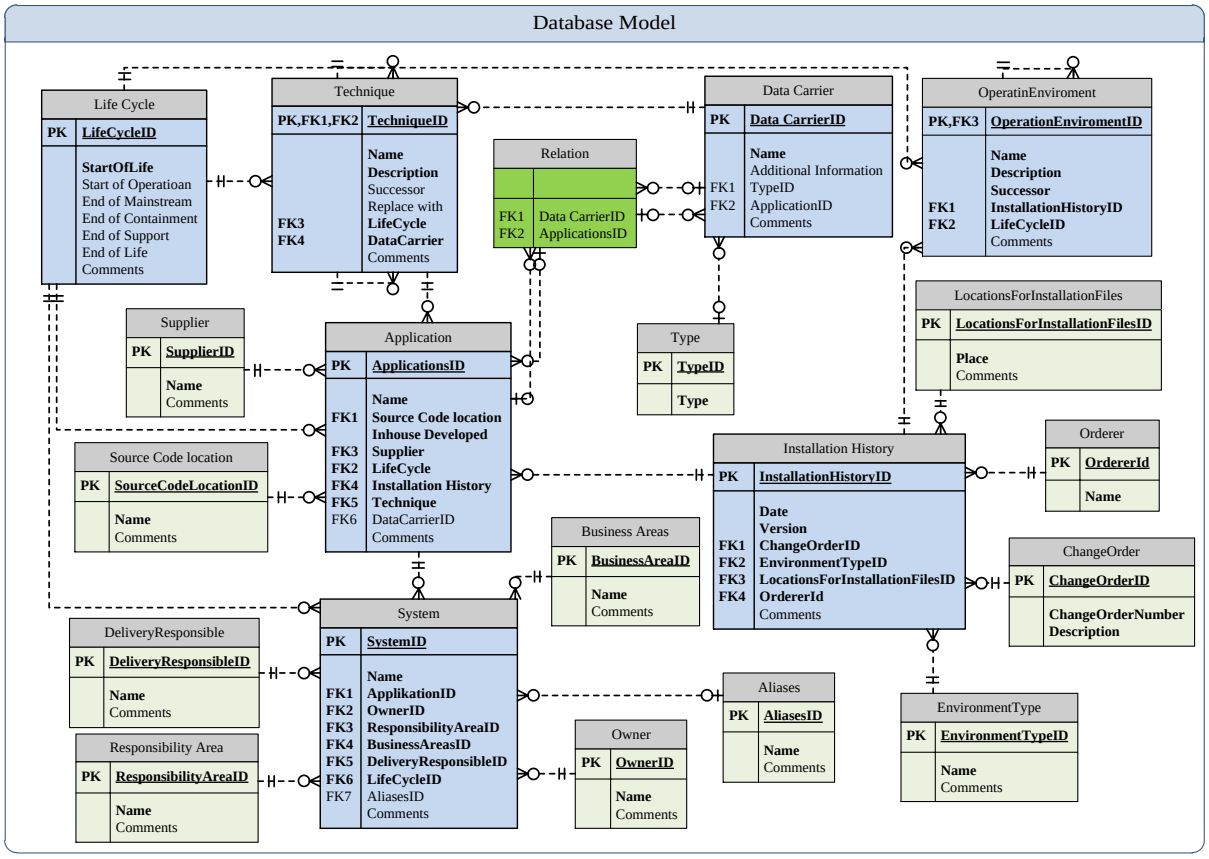


Figure 8: First layout: Database Model

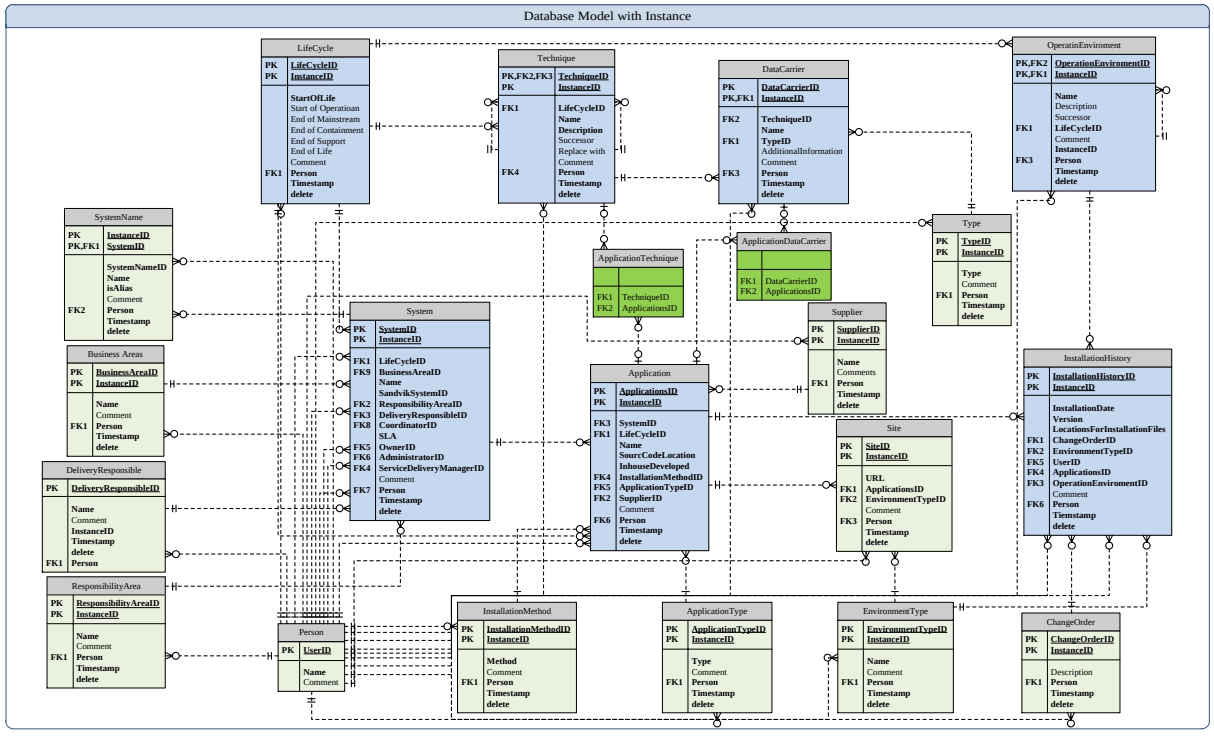


Figure 9: Database Model with Instance

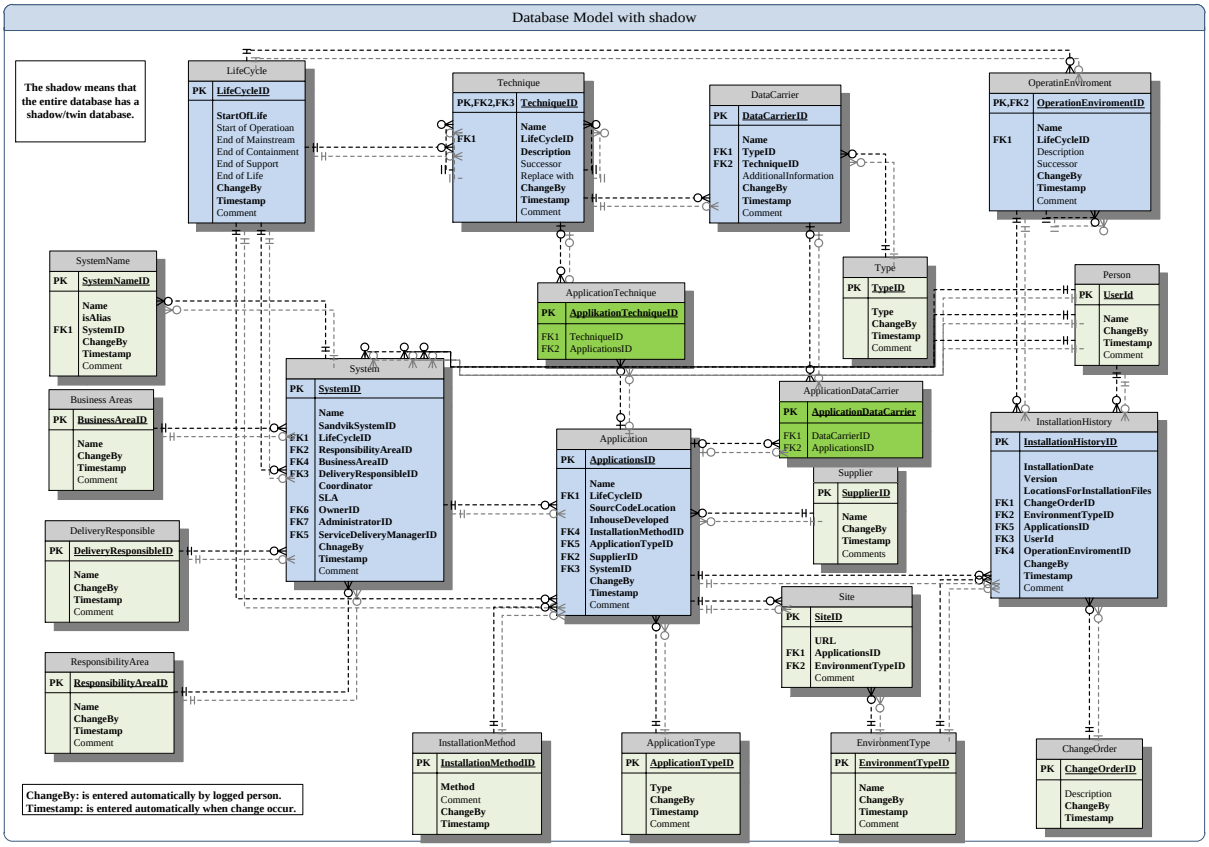


Figure 10: Database Model with shadow

The first column shows which key, *PK* – Primary Key or *FK* – Foreign Key. The numbers on the keys (FK1, FK2…) is to separate them.
The third column in each table is Data types.

Data type	Description
Datetime	Date and time
Int	Integer is a number without decimal
nvarchar(x)	A variable-length Unicode String data, *x* defines the length of the String
Bit	Basic unit that contains only 2 values, e.g. 0 and 1 or logical value true/false, yes/no

6 Analysis and Discussion

In this report my task was to create architecture for a *life cycle system* suitable for holding the information needed by an application management organization.
Collecting the information that was needed was conducted through interviews and a workshop. It went well on the interviews because I had prepared myself before with questions. I was less prepared for the workshop due to short notice and that I have never led a workshop, but it went well because the participants were accustomed with workshops.
To analyze the information and create an information model took some time and lots of meetings before we could come to a conclusion.

Creating the database model was generally smoother because the hardest work was done in the creation of the information model. The hardest part was how to solve the record of history of who made changes in the system.
I feel that I have accomplished my task plus I will have time to develop a Proof of Concept. However, I do not have time to documents this, but I have written down how I´m going to do and I have created some possible examples of reports.
I think this system will serve its purpose in Windows Development and also in other departments.

7 Reference

[1] Sandvik AB. http://sandvik.com/en/about-sandvik (accessed June - 2013)

[2] Sandvik Construction *http://www.construction.sandvik.com/* (accessed June - 2013)

[3] Sandvik Machining Solutions (SMS) *http://www.sandvik.coromant.com/en-gb/pages/default.aspx* (accessed June - 2013)

[4] Sandvik Materials Solutions (SMT) *http://www.smt.sandvik.com/* (accessed June - 2013)

[5] Sandvik Mining Source *http://www.mining.sandvik.com/* (accessed June - 2013)

[6] Sandvik Venture http://www.hardmaterials.sandvik.com/ (accessed June - 2013)

[7] Sandvik Venture http://www.processsystems.sandvik.com/ (accessed June - 2013)

[8] Sandvik IT http://www.sandbackapark.com/pages.asp?PageID=416&MenuID=345 (accessed June - 2013)

[9] Intranet of Sandvik AB (accessed June - 2013)

[10] Vad är en databasmodell? http://home.swipnet.se/~w-69951/dbmodeller.htm

[11] What is Kanban? http://www.kanbanblog.com/explained/

[12] Scrum (Software development) http://en.wikipedia.org/wiki/Scrum_(software_development)

8 Appendix

8.1 Appendix A, Simple plan

The writing of documents and reports will be done on going during the specified time period. Invites and bookings for the presentation on Sandvik AB and HiG should be done latest week 2.

W.1 – 3:

- Collect data.
- Analyze data.
- Create Information model for the database.
    - Business Analysis
    - Requirement an problem analyze
    - Establish Main purpose.

W.4 – 5:

- Create Database Model
    - Establish functionality
    - Expected effects
    - Goal-settings

W.6

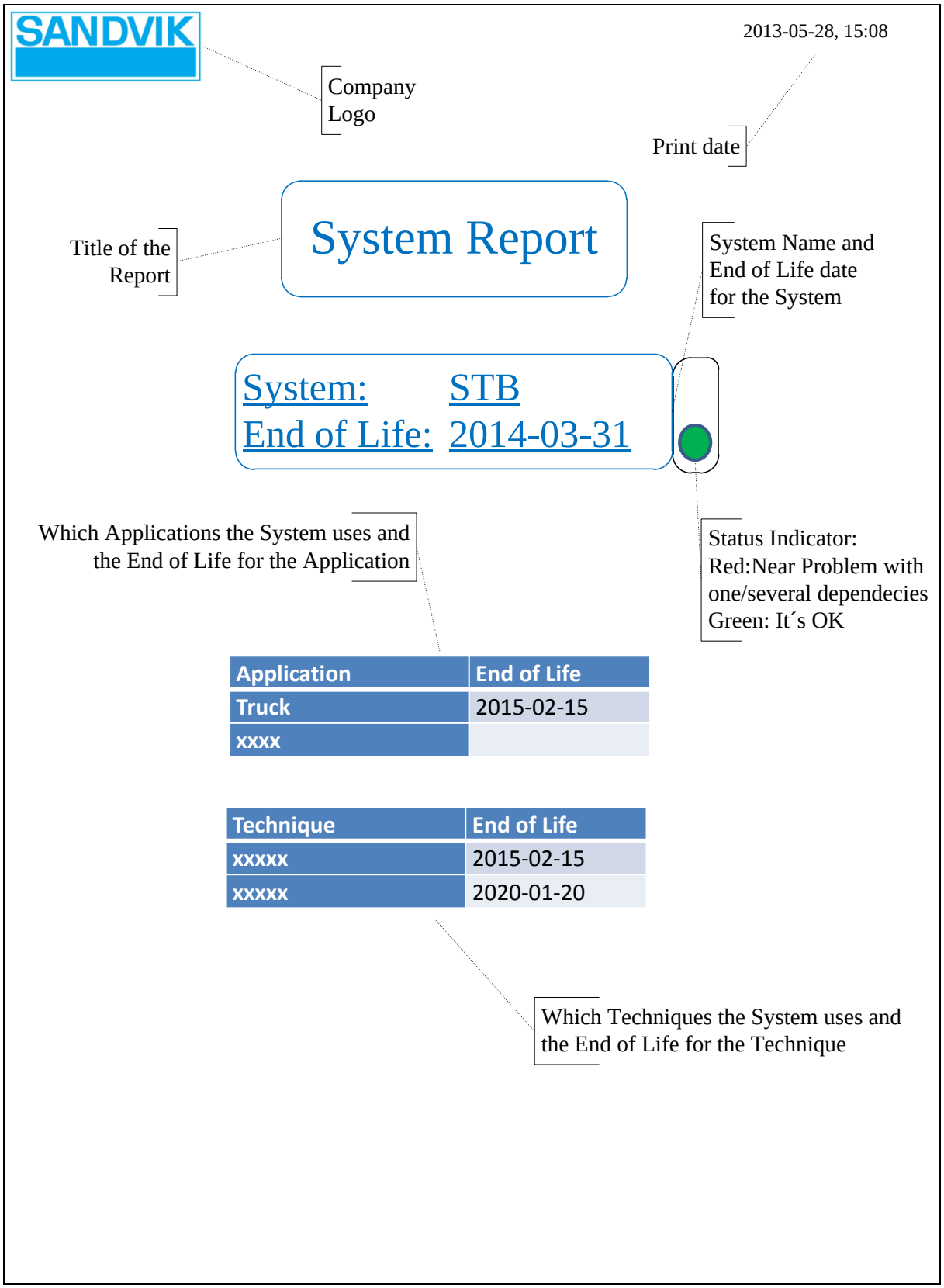
- Results and summary
- Create a presentation
- Finish report
- If there is time – create a "proof of concept"

v.7

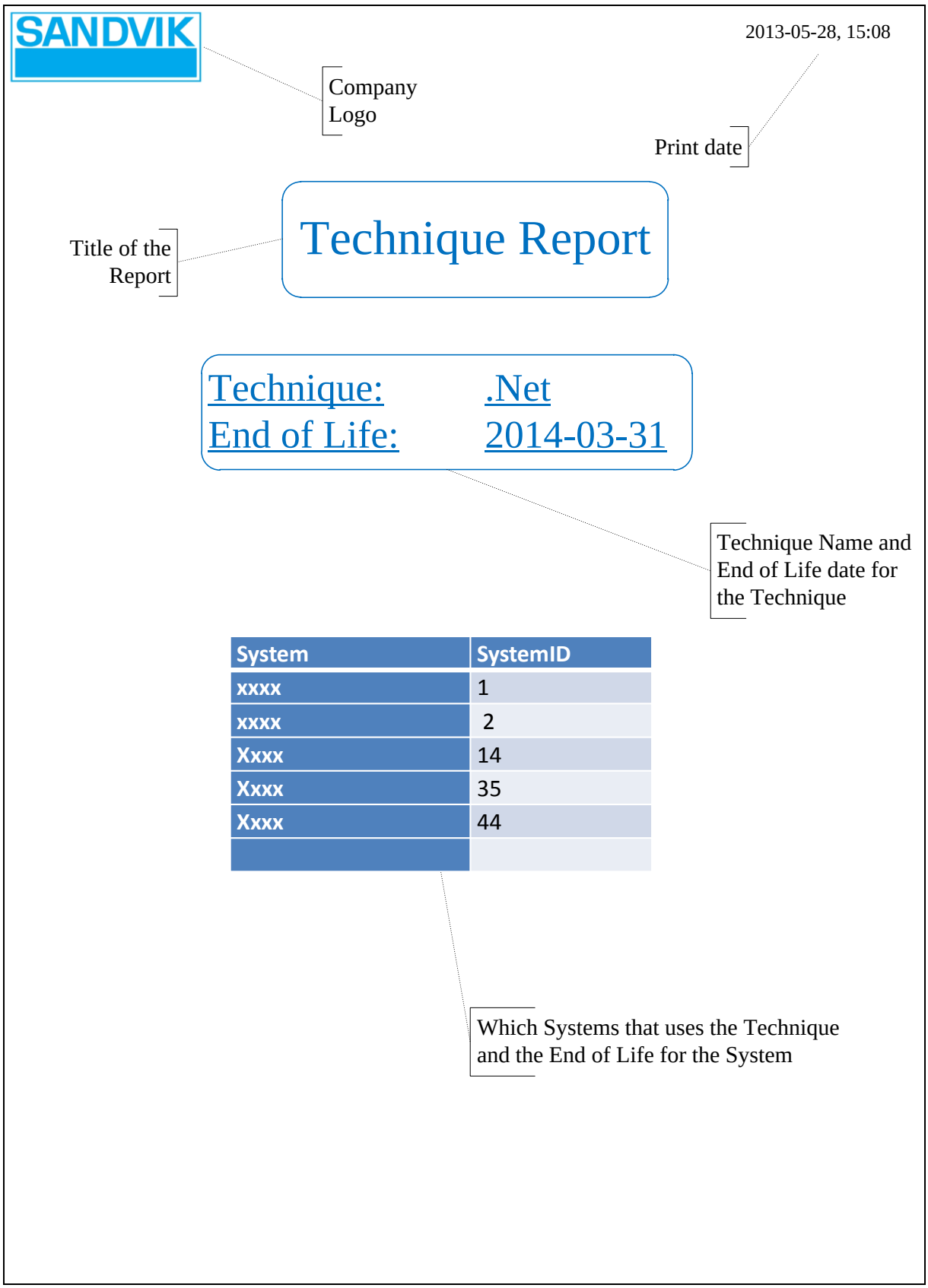
- Presentation on Sandvik AB.
- Presentation on HiG

Planned by: Izzet Canciger

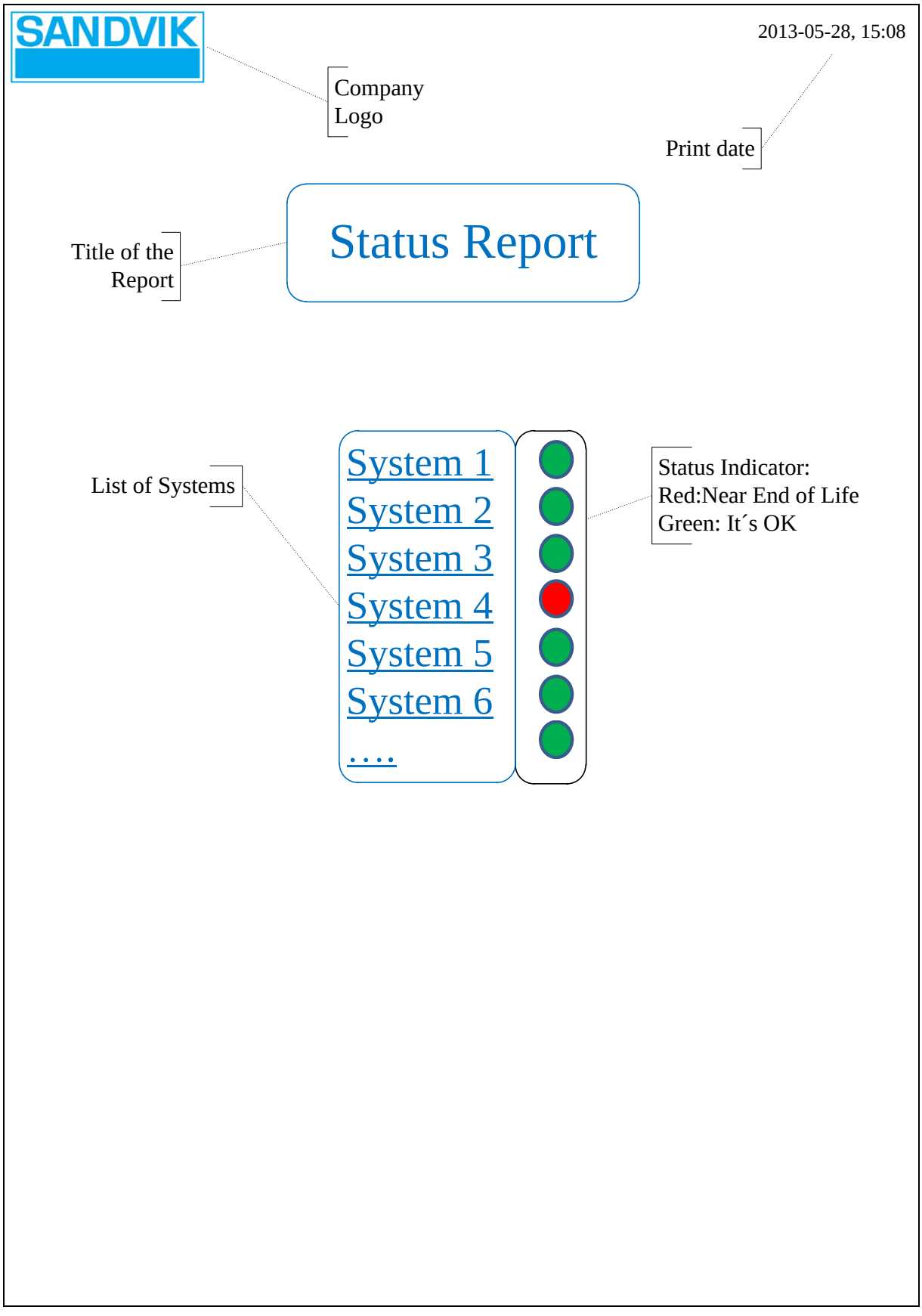
8.2 Appendix B, Example of System Report for Proof of Concept



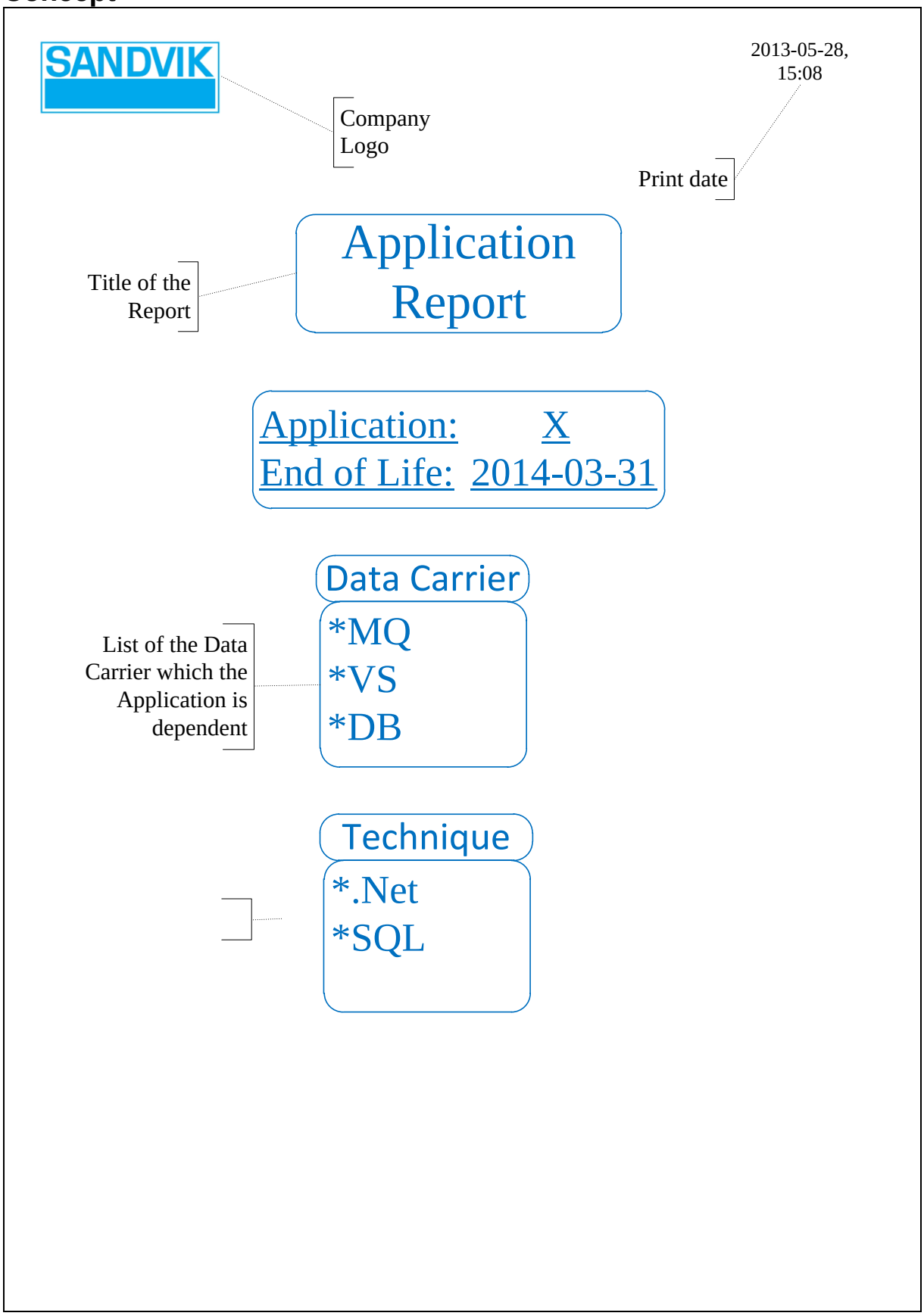
8.3 Appendix C, Example of Technique Report for Proof of Concept



8.4 Appendix D, Example of Status list for Proof of Concept



8.5 Appendix E, Example Application Report for Proof of Concept



8.6 Appendix F, Attributes description

Life Cycle

Attributes	Description
Start of Life	Birthdate of the product
Start of Operational	Start date of Operational state of the product
End of Mainstream	End date of mainstream period of the product
End of Containment	End date of Containment period of the product
End of Support	End date of Support period of the product
End of Life	Closure date for the product, not available after this date
Comments	Free text

Operating Environment

Attributes	Description
Name	Name of the *Operating Environment*, e.g. WDE2, WDE3, SNCG3
Description	Description of the *Operating Environment*
Successor	Next generation of the *Operating Environment*
Comments	Free text

Technique

Attributes	Description
Name	Name of the *Technique*, e.g. .Net 3.0, VB6
Description	Description of the *Technique*
Successor	Next generation of the current *Technique*
Replace with	Which Version/ Product that will replace current
Comments	Free text

System

Attributes	Description
Name	Name of the System, e.g. STB
Owner	Owner of the System
Responsibility Area	Which SDM (Service Delivery Manager) area, e.g. SDM Production, SDM Manufacturing…
Business Areas	e.g. SMT, SMS…
Delivery Responsible	Organization or External supplier that delivers services for the system
Aliases	Names used by others
Comments	Free text

Application

Attributes	Description
Name	Name of the *Application*, e.g. STB Web, STB Lift, STB Web service.
Source Code location	Where the Source Code is.
In house Developed	Own development
Supplier	Creator/Author of the *Application*. Internal or External
Comments	Free text

Installation History

Attributes	Description
Date	Which *date* the product is installed.

Version	Which *version* the product.
Environment Type	Which *Environment Type* the product is in, e.g. development, test, and production.
Order requester	Who ordered the *Installation?*
Change order	The *Change order* number.
Locations for installations files	The *locations for installations files*
Comments	Free text

Data Carriers

Attributes	**Description**
Name	Name of the *Item*
Type	Which Type the *Item* is e.g. MQ-queue, Web services, iBridge etc.
Additional information	Additional information e.g. server name, server manager etc.
Comments	Free text

8.7 **Appendix G,** Attributes description (that have been added)

For all tables

Attributes	Description
ChangeBy	UserID retrieved from the user that has done the change
Timestamp	Date and time are entered when change was made
xID	The ID number for the corresponding tables

System

Attributes	Description
SandvikSystemID	The ID number in the Sandvik System

Application

Attributes	Description
Name	Name of the *Application*, e.g. STB Web, STB Lift, STB Web service.
Source Code location	Where the source code is.
In house Developed	Flag indicates if the code is developed in house or not
Supplier	Creator/Author of the *Application.* Internal or External
Comments	Free text

Installation History

Attributes	Description
Date	Which *date* the product is installed.
Version	Which *version* of the product.
Environment Type	Which *Environment Type* the product is in, e.g. development, test, production….
Order requester	Who ordered the *Installation?*
Change order	The *Change order* number.
Locations for installations files	The *locations for installations files*
Comments	Free text

Data Carriers

Attributes	Description
Name	Name of the *Item*
Type	Which Type the *Item* is e.g. MQ-queue, Web services, iBridge etc.
Additional information	Additional information e.g. server name, server manager etc.
Comments	Free text